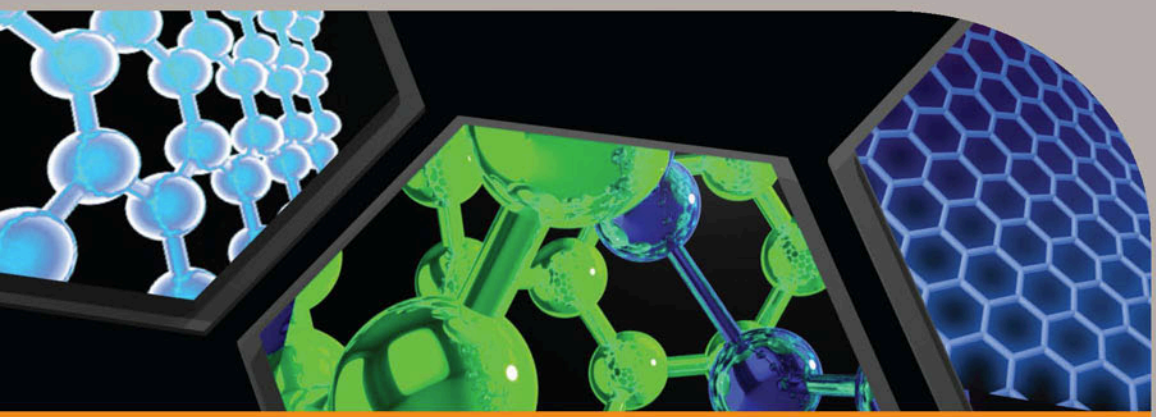
Materials, Circuits and Devices Series 30



Nano-CMOS and Post-CMOS Electronics: Circuits and Design

Edited by
Saraju P. Mohanty
and Ashok Srivastava

VISIT…

MATERIALS, CIRCUITS & DEVICES SERIES 30

Nano-CMOS and Post-CMOS Electronics: Circuits and Design

Other volumes in this series:

Volume 2 **Analogue IC Design: The current-mode approach** C. Toumazou, F.J. Lidgey and D.G. Haigh (Editors)
Volume 3 **Analogue-Digital ASICs: Circuit techniques, design tools and applications** R.S. Soin, F. Maloberti and J. France (Editors)
Volume 4 **Algorithmic and Knowledge-based CAD for VLSI** G.E. Taylor and G. Russell (Editors)
Volume 5 **Switched Currents: An analogue technique for digital technology** C. Toumazou, J.B.C. Hughes and N.C. Battersby (Editors)
Volume 6 **High-frequency Circuit Engineering** F. Nibler *et al.*
Volume 8 **Low-power High-frequency Microelectronics: A unified approach** G. Machado (Editor)
Volume 9 **VLSI Testing: Digital and mixed analogue/digital techniques** S.L. Hurst
Volume 10 **Distributed Feedback Semiconductor Lasers** J.E. Carroll, J.E.A. Whiteaway and R.G.S. Plumb
Volume 11 **Selected Topics in Advanced Solid State and Fibre Optic Sensors** S.M. Vaezi-Nejad (Editor)
Volume 12 **Strained Silicon Heterostructures: Materials and devices** C.K. Maiti, N.B. Chakrabarti and S.K. Ray
Volume 13 **RFIC and MMIC Design and Technology** I.D. Robertson and S. Lucyzyn (Editors)
Volume 14 **Design of High Frequency Integrated Analogue Filters** Y. Sun (Editor)
Volume 15 **Foundations of Digital Signal Processing: Theory, algorithms and hardware design** P. Gaydecki
Volume 16 **Wireless Communications Circuits and Systems** Y. Sun (Editor)
Volume 17 **The Switching Function: Analysis of power electronic circuits** C. Marouchos
Volume 18 **System on Chip: Next generation electronics** B. Al-Hashimi (Editor)
Volume 19 **Test and Diagnosis of Analogue, Mixed-signal and RF Integrated Circuits: The system on chip approach** Y. Sun (Editor)
Volume 20 **Low Power and Low Voltage Circuit Design with the FGMOS Transistor** E. Rodriguez-Villegas
Volume 21 **Technology Computer Aided Design for Si, SiGe and GaAs Integrated Circuits** C.K. Maiti and G.A. Armstrong
Volume 22 **Nanotechnologies** M. Wautelet *et al.*
Volume 23 **Understandable Electric Circuits** M. Wang
Volume 24 **Fundamentals of Electromagnetic Levitation: Engineering sustainability through efficiency** A.J. Sangster
Volume 29 **Nano-CMOS and Post- CMOS Electronics: Devices and Modelling** Saraju P. Mohanty and Ashok Srivastava

Nano-CMOS and Post-CMOS Electronics: Circuits and Design

Edited by
Saraju P. Mohanty
and Ashok Srivastava

The Institution of Engineering and Technology

Published by The Institution of Engineering and Technology, London, United Kingdom

The Institution of Engineering and Technology is registered as a Charity in England & Wales (no. 211014) and Scotland (no. SC038698).

© The Institution of Engineering and Technology 2016

First published 2016

This publication is copyright under the Berne Convention and the Universal Copyright Convention. All rights reserved. Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may be reproduced, stored or transmitted, in any form or by any means, only with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publisher at the undermentioned address:

The Institution of Engineering and Technology
Michael Faraday House
Six Hills Way, Stevenage
Herts, SG1 2AY, United Kingdom

www.theiet.org

While the authors and publisher believe that the information and guidance given in this work are correct, all parties must rely upon their own skill and judgement when making use of them. Neither the authors nor publisher assumes any liability to anyone for any loss or damage caused by any error or omission in the work, whether such an error or omission is the result of negligence or any other cause. Any and all such liability is disclaimed.

The moral rights of the authors to be identified as authors of this work have been asserted by them in accordance with the Copyright, Designs and Patents Act 1988.

British Library Cataloguing in Publication Data
A catalogue record for this product is available from the British Library

ISBN 978-1-84919-999-5 (hardback)
ISBN 978-1-78561-000-4 (PDF)

Typeset in India by MPS Limited
Printed in the UK by CPI Group (UK) Ltd, Croydon

Contents

Preface **xiii**

1 Self-healing analog/RF circuits **1**
1.1 Introduction 1
1.2 Indirect performance sensing 3
1.3 Pre-silicon indirect sensor modeling via SR 4
1.3.1 L_0-norm regularization 5
1.3.2 L_1-norm regularization 7
1.3.3 Accuracy of L_1-norm regularization 9
1.4 Post-silicon indirect sensor calibration via Bayesian model fusion 11
1.4.1 Prior knowledge definition 12
1.4.2 MAP estimation 14
1.5 On-chip self-healing flow 17
1.6 Case study 20
1.6.1 25 GHz differential Colpitts VCO 20
1.6.2 60 GHz LNA 26
1.7 Conclusion 32
References 32

2 On-chip gate delay variability measurement in scaled technology node **35**
2.1 Introduction 35
2.2 Classification of variability 36
2.3 Sources of variability 38
2.3.1 Random dopant fluctuations 38
2.3.2 Line edge roughness 38
2.3.3 Oxide thickness variation 39
2.4 Related work on variability measurement 39
2.4.1 Gate delay variability 39
2.4.2 Rise and fall gate delay variability 41
2.5 Gate delay measurement using reconfigurable ring oscillator 42
2.5.1 Gate delay measurement cell 42
2.5.2 Reconfigurable ring oscillator structure 43
2.5.3 Measured results 48
2.5.4 Poly-pitch effect 49
2.5.5 Length of diffusion effect 51
2.5.6 Delay variation due to layout orientation 52

2.5.7 Delay variation due to supply voltage 52
2.5.8 Measured accuracy of the delay measurement 53
2.5.9 Comparison with other works 55
2.6 Measurement of rise and fall delays using standard RO 56
2.7 Rise and fall gate delay measurement using RRO 57
2.7.1 Gate delay measurement cell 57
2.7.2 Rise and fall delays of non-inverting gate 58
2.7.3 Rise and fall delays of inverting gate 60
2.8 Test chip and measurement results 61
2.8.1 Measurement of duty cycle 61
2.9 Measured results 62
2.9.1 Impact of body-bias 63
2.9.2 Impact of supply voltage 65
2.9.3 Measurement accuracy 65
2.9.4 Comparison with the existing techniques 65
2.10 Summary and conclusions 66
References 67

3 Nanoscale FinFET devices for PVT-aware SRAM 71
3.1 Introduction 71
3.2 Nanoscale FinFET devices 72
3.2.1 Bulk FinFET 73
3.2.2 SOI FinFET 75
3.3 FinFET-based SRAM topologies 79
3.3.1 IG-FinFET-based 6T SRAM 80
3.3.2 Back-gate bias IG-FinFET-based 6T SRAM 82
3.3.3 IG-FinFET-based PPN 10T SRAM 83
3.3.4 Stability analysis 88
3.4 FinFET-based SRAM design challenges 91
3.5 PVT-aware SRAM design 92
3.5.1 PVT mitigation techniques 93
3.5.2 PVT-aware SRAM designs 95
3.5.3 Stability analysis 103
3.6 Conclusion 106
References 106

4 Data stability and write ability enhancement techniques for FinFET SRAM circuits 113
4.1 Introduction 113
4.2 Six-FinFET SRAM cells 114
4.2.1 Conventional six-FinFET SRAM cell 114
4.2.2 Independent-gate FinFET SRAM cell 116
4.2.3 SRAM cell with asymmetrically overlap/underlap engineered FinFETs 117

4.2.4 Hybrid SRAM cell with asymmetrically overlapped/underlapped bitline access transistors 120
4.2.5 SRAM cell with asymmetrically gate-underlapped transistors 121
4.2.6 Single-ended read SRAM cell with underlap engineered symmetrical-FinFETs 125
4.3 Fabrication and SRAM cell area comparison 127
4.4 Case study: 8KBit memory arrays designed with different SRAM cells 128
4.4.1 Read static noise margin 128
4.4.2 Hold static noise margin 129
4.4.3 Write voltage margin 130
4.4.4 Data access speed 131
4.4.5 Leakage power consumption 132
4.5 Variations of underlap (overlap) lengths due to process imperfections 133
4.6 Conclusions 138
References 138

5 Low-leakage techniques for nanoscale CMOS circuits 141
5.1 Introduction 141
5.2 Device scaling 142
5.2.1 Constant voltage scaling 144
5.2.2 Constant field scaling 144
5.2.3 Generalized scaling 145
5.3 Power dissipation 145
5.3.1 Leakage power dissipation 145
5.3.2 Leakage current components 146
5.4 Issue of leakage current 148
5.5 Variability issues and aware design 148
5.6 Leakage reduction techniques 150
5.6.1 MTCMOS technique 150
5.6.2 Forced stack technique 152
5.6.3 Dual threshold CMOS (DTCMOS) technique 154
5.6.4 SCCMOS (super cut-off CMOS) technique 154
5.6.5 Leakage feedback technique 156
5.6.6 Variable threshold CMOS (VTCMOS) technique 157
5.6.7 LECTOR technique 158
5.6.8 Sleepy stack technique 158
5.6.9 Sleepy keeper technique 159
5.6.10 VCLEARIT technique 160
5.6.11 GALEOR technique 161
5.7 Leakage analysis 162
5.8 Conclusion 167
References 167

6 Thermal effects in carbon nanotube VLSI interconnects 173
6.1 Introduction 173
6.2 Present status of VLSI interconnect 174
6.3 Survey of CNT-based interconnects 175
6.4 Electrical properties 176
6.4.1 Equivalent resistance (R_{eqv}) 177
6.4.2 Equivalent inductance (L_{eqv}) 180
6.4.3 Equivalent capacitance (C_{eqv}) 181
6.4.4 Effective mean free path (λ_{eff}) 182
6.4.5 Equivalent circuit 184
6.5 Thermal properties 185
6.5.1 Thermal properties of SWCNTs 185
6.5.2 Thermal properties of SWCNT bundle 187
6.5.3 Thermal properties of MWCNT 189
6.5.4 Iterative scheme for R and T 190
6.5.5 Temperature profiling inside the interconnect 191
6.5.6 Performances in terms of S-parameters 193
6.6 Conclusion 196
References 196

7 Lumped electro-thermal modeling and analysis of carbon nanotube interconnects 201
7.1 Introduction 201
7.2 Electrical modeling of CNTs 202
7.3 Thermal modeling for CNTs 205
7.4 Conclusion 216
References 217

8 High-level synthesis of digital integrated circuits in the nanoscale mobile electronics era 219
8.1 Introduction 219
8.2 Fundamentals on high level synthesis 222
8.2.1 Overview on HLS design process 222
8.2.2 Need for HLS 224
8.2.3 Scheduling algorithms 225
8.2.4 Allocation and binding 228
8.3 Power, energy, or leakage aware HLS for nanoscale ICs 229
8.3.1 Selected power, energy, or leakage aware HLS methods 229
8.3.2 Effects of loop manipulation on power and delay of the design 233
8.3.3 Other design space exploration approaches during HLS 239
8.4 Bio/nature-inspired algorithms for DSE framework 241
8.4.1 Selected bio/nature-inspired approaches 241
8.4.2 A BFOA-exploration process 243
8.4.3 Encoding/initialization of the datapath bacterium 244

8.4.4 Encoding of the auxiliary bacterium 245
8.4.5 Proposed movement of bacterium 246
8.4.6 Models for metric 248
8.4.7 Results of the BFOA-exploration process 248
8.5 HLS approaches for secure information processing 252
8.5.1 Related work 252
8.5.2 Exploration process of hardware Trojan secured datapath: security against untrusted third party digital IPs 253
8.5.3 Results of exploration process of hardware Trojan secured datapath 258
8.6 Selected tools available for HLS 258
8.6.1 Selected commercial tools for HLS 259
8.6.2 Selected free HLS tools 260
8.7 Conclusion and future directions of HLS 260
References 261

9 SPICEless RTL design optimization of nanoelectronic digital integrated circuits 267
9.1 Introduction 267
9.2 The concept of SPICEless RTL optimization during HLS 271
9.3 The issues in RTL optimization of power dissipation in digital circuits 272
9.4 Power optimization at RTL: state-of-the-art 274
9.4.1 Existing methods for RTL power optimization 274
9.4.2 Multiple oxide thickness technology for gate-oxide leakage optimization 276
9.5 A specific SPICEless RTL optimization approach 278
9.5.1 The overall RTL optimization flow 278
9.5.2 Objective function for RTL optimization 279
9.5.3 A specific heuristic algorithm for RTL optimization 281
9.6 SPICEless characterization of the RTL component library 285
9.6.1 Gate-oxide leakage modeling 287
9.6.2 Propagation delay modeling 289
9.6.3 Analytical modeling of RTL components 291
9.7 Experimental results for the specific RTL optimization 293
9.8 Conclusions and future directions of research 298
Acknowledgments 299
References 299

10 Green on-chip inductors for three-dimensional integrated circuits: concepts, algorithms and applications 305
10.1 Introduction 305
10.2 Effect of various parameters of an on-chip inductor 307
10.2.1 Impact of process parameters 309
10.2.2 Design parameters 313

10.3 Low-frequency applications 316
10.3.1 DC–DC converter design 316
10.3.2 Resonant clocking implementation 323
10.4 Micro-channel shielding 326
10.5 Summary and conclusions 334
References 334

11 3D NoC: a promising alternative for tomorrow's nanosystem design 337
11.1 Introduction 337
11.1.1 NoC basics 338
11.1.2 Transition towards 3D 338
11.2 Design challenges in 3D NoC 340
11.2.1 Design challenges 341
11.2.2 Macro-architecture 341
11.2.3 Emerging technological challenges 345
11.3 Performance centric design of 3D NoCs 347
11.3.1 Interconnection topology development 347
11.3.2 Routing policy 347
11.3.3 Flow control mechanism 349
11.4 Architectural optimization of 3D NoCs 349
11.4.1 Router architecture 349
11.4.2 Network interface controller 350
11.4.3 Interconnection 351
11.4.4 Memory 352
11.5 Thermal-aware design 352
11.6 Photonic 3D NoC 353
11.6.1 Photonic interconnect for manycore ICs 353
11.6.2 Multi-dimensional design issues in 3D PNoC 357
11.7 Wireless 3D NoC 358
11.7.1 Low-latency-based wireless 3D NoCs 358
11.7.2 Inductive coupling interconnected application-specific 3D NoC 359
11.7.3 Reconfigurable hybrid 3D wireless NoC 360
11.8 3D NoC simulators 361
11.8.1 NoC simulation 361
11.9 Reliability and fault tolerance in 3D NoCs 364
11.10 Conclusion 366
References 366

12 A new paradigm towards performance centric computation beyond CMOS: DNA computing 379
12.1 Introduction 379
12.1.1 DNA structure 380
12.1.2 Operations on DNA solutions 381

12.1.3 How DNA computers work? Power of DNA computer 386
12.1.4 History of DNA computing 386
12.2 DNA computing models 387
12.2.1 Adleman–Lipton model 388
12.2.2 Sticker model 393
12.3 Performing arithmetic and logic operations using DNA 397
12.3.1 AND operation 398
12.3.2 OR operation 399
12.3.3 XOR operation 400
12.3.4 NOT operation 400
12.3.5 Comparator 400
12.4 Implementing data structures using DNA 402
12.4.1 Stack and queue using DNA 402
12.4.2 List using DNA 403
12.4.3 Map using DNA 404
12.5 Conclusion 405
References 405

Index **409**

Preface

Overall population growth in the urban areas has been posing critical challenges for social life, healthcare, and even basic food supplies. As a mitigation of this problem, future smart cities are envisioned to have many smart frameworks or systems including smart technology, smart healthcare, smart grids, smart transportation, smart buildings, smart communication, and smart information technology. For example, a smart health care system such as the body-area network (BAN) is able to provide quality health care to patients even when doctors cannot be present but are available remotely. A smart transport system is able to provide real-time locations of the entities in transportation system network. Generally speaking, the information and communication technology (ICT) is the core of such smart systems from a small- to a large-scale implementation. A combination of hardware and software can implement such smart systems through the realization of the ICT. But software does need hardware as a base to be executed on. Hardware in the smart system can be quite diverse, such as can be sensors of any type, analog integrated circuits (ICs), digital ICs, or even mixed-signal ICs. The hardware is designed by design engineers at various levels of abstractions depending on their nature whether, analog or digital. The overall design process is, however, based on a specific process technology to manufacture ICs and systems. Current chip manufacturing processes use nanometer-scale CMOS (nano-CMOS), and post-CMOS technologies which is generally known as nanoelectronic technology.

For efficient realization of nanoscale device-based systems, with the development of emerging nanoscale devices, their design and manufacturing processes need to develop and mature. Detailed discussion of these issues, such as power dissipation, leakage power, and information security, and their corresponding solutions, such as different circuits and design flows, are lacking in existing texts and existing curricula in academia. Most importantly the design engineers' tasks have been severely complicated due to the emergence of these issues which has led to longer design cycle time. Hence, yield loss and as a consequence high chip costs are common, and the overall impact is the increased cost of consumer electronics which are used in day-to-day life. There is a large gap between the skill of design engineers and understanding of these devices and their integration in design methodologies. However, existing books are typically based on traditional CMOS devices and do not cater the needs of new circuit topology as well as circuit and system design aspects using post-CMOS or nanoelectronic devices. As a consequence, existing books do not train future generation engineers in emerging nanoscale device-based electronics, circuits and systems, and hence do not catalyze the growth of nanoelectronics. The traditional literatures do not serve the expectations of the emerging

nanotechnology industry; however, this book will meet the demand of training future generation of engineers in emerging nanoelectronic circuits and systems. As a consequence, existing books do not educate and train engineers in emerging nano-devices and do not catalyze the growth of nanoelectronics. The traditional literature does not fulfill the expectation of the emerging nanoelectronic design and manufacturing industry. This book will meet the demand of educating and training engineers in nanoelectronics. For the device level modeling a book titled "CMOS and Post-CMOS Electronic Device Scaling: Devices and Modeling" has been already released.

A natural progression of the device level modeling is the IC or system design using the device level information. For efficient design of nanoelectronics circuits and systems and design space exploration, understanding of the devices and design methodologies are crucial. Through the help of new methodologies and electronic design automation (EDA) tools, one can train students and researchers of many engineering disciplines in nanoelectronics starting from devices/circuits to architecture/systems. The nanoelectronics challenges are the higher levels of abstraction such as logic, and system, different from the device level modeling. The issues are also different for analog design engineers as compared to a digital design engineer. At the circuit and system levels, various challenges including process variations, strict energy budget, interconnect effects, thermal effects, yield issue, and time-to-market constrains may arise. However, existing text books do not address these which slow down the training of manpower and hence growth of nanoelectronics. The current book titled "CMOS and Post-CMOS Electronic Device Scaling: Circuits and Designs" has been presented to fulfill this urgent need.

In this book, the chapter titled "Self-Healing Analog and Radio Frequency Circuits" by Shupeng Sun, et al. discusses process variation in nanoscale analog and radio frequency ICs. It discusses an important dynamic method called "self-healing" to actively monitor the post-manufacturing circuit performance metrics and then adaptively adjust a number of tuning knobs to meet the given performance specifications. In the chapter titled "On-Chip Gate Delay Variability Measurement in Scaled Technology Node" by Das, Amrutur, and Onodera, process variations in digital ICs are presented. The focus is on the impact of process variations in digital ICs with propagation delay as the target characteristic. The chapter titled "Nanoscale FinFET Devices for PVT-Aware SRAM" by Sharma, Pattanaik, and Yadav focuses on the static random access memory (SRAM) chips using the FinFET. It presents a comprehensive discussion on variability including process variation, voltage variation, and temperature variation. The chapter titled "Data Stability and Write Ability Enhancement Techniques for FinFET SRAM Circuits" by Salahuddin and Kursun further discusses the FinFET-based SRAM for improving stability. Specifically, several alternative FinFET memory design techniques are presented in this chapter for achieving stronger data stability during the read operation and wider voltage margin during the write operation. The chapter titled "Low-Leakage Techniques for Nanoscale CMOS Circuits" by Pattanaik and Sharma elaborates another important issue of nanoscale CMOS circuits, called leakage power dissipation. This chapter discusses selected important

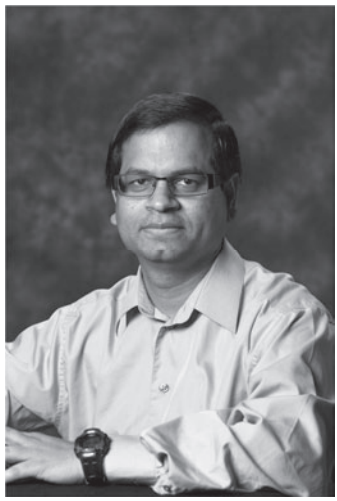
techniques for leakage power reduction in nanoscale CMOS ICs. The chapter titled "Thermal Effects in Carbon Nanotube VLSI Interconnects" by Srivastava and Mohsin presents thermal effect in interconnects of the ICs. In a paradigm shift, not traditional metal interconnects, but carbon nanotube (CNT)-based interconnects are considered as a possible replacement for Copper. The chapter titled "Lumped Electro-Thermal Modeling and Analysis of Carbon Nanotube Interconnects" by Todri-Sanial presents detailed electro-thermal analyses of horizontally aligned CNTs and presents their performance and voltage drop.

The chapter titled "High-Level Synthesis of Digital Circuits in the Nanoscale Mobile Electronics Era" by Sengupta and Mohanty presents detailed discussions of high-level synthesis (HLS) techniques which can generate digital ICs. This chapter also discusses the HLS technique that can generate trusted digital ICs as trust/security of electronic systems that are used in day-to-day life. The chapter titled "SPICE-less RTL Design Optimization of Nanoelectronic Digital Integrated Circuits" by Kougianos and Mohanty presents HLS methods for leakage-optimal digital IC design exploration. Specifically, a paradigm shift approach is presented in which the complete HLS flow is performed without use of any EDA tool. The chapter titled "Green On-chip Inductors for Three-Dimensional Integrated Circuits: Concepts, Algorithms and Applications" by Tida, Zhuo, Jain, Krishnamurthy, and Shi discusses three-dimensional ICs (3D ICs) as compared to the planner ICs of the previous chapters. This chapter specifically discusses practical approaches to through-silicon-vias (TSV) inductors which constitutes the vertical signal, power, and thermal paths which are very critical for 3D ICs. The chapter titled "3D NoC: A Promising Alternative for Tomorrow's Nanosystem Design" by Ghosal, Das, Poddar, Rahaman, and Bose discusses different design challenges, available technologies, design and performance issues and parametric measurement of such nanoscale systems, emerging cutting-edge technologies, and possible future directions in designing 3D NoC-based nanosystems. The chapter titled "A New Paradigm towards Performance Centric Computation beyond CMOS: DNA Computing" by Ghosal, Sarkar, and Chatterjee presents an introduction to structure of DNA and how DNA computing works and several aspects of DNA computing.

This book will address many of the nanoelectronic circuit/system level issues as well as integrated circuit and nano-system design methods for efficient design exploration of nano-CMOS- and post-CMOS-based systems. Special features of the book include the following:

(1) Coverage of various circuit and system level issues and solutions.
(2) Coverage of both analog and digital nanoelectronics systems.
(3) Coverage of issues and solutions for both device and interconnects.
(4) Coverage of power, thermal, and variability issues and solutions.
(5) Coverage of emerging platforms such as 3D IC and DNA computing.
(6) Coverage of key issues, challenges, and solutions of nanoelectronic system design challenges that the industry is striving to address.
(7) Coverage of design methods accounting for nanoscale issues and challenges.

This book can serve as reference for graduate students (Ph.D./M.S.), researchers, and practicing engineers. Master students and senior undergraduate students will benefit from the contents of this book. Students of various disciplines such as Computer Engineering, Electrical Engineering, Computer Science with VLSI Design, VLSI Computer-Aided Design (CAD), and Embedded System Design will be benefited by this book.



Saraju P. Mohanty
Professor, University of North Texas, USA.
saraju.mohanty@unt.edu

Ashok Srivastava
Professor, Louisiana State University, USA.
eesriv@lsu.edu

Chapter 1

Self-healing analog/RF circuits

Shupeng Sun[1], *Fa Wang*[1], *Soner Yaldiz*[1], *Xin Li*[1], *Lawrence Pileggi*[1], *Arun Natarajan*[2], *Mark Ferriss*[2], *Jean-Olivier Plouchart*[2], *Bodhisatwa Sadhu*[2], *Ben Parker*[2], *Alberto Valdes-Garcia*[2], *Mihai A. T. Sanduleanu*[2], *Jose Tierno*[2] *and Daniel Friedman*[2]

Process variation is the most critical issue for the nanoscale analog and radio-frequency integrated circuits (ICs). There are many traditional techniques to mitigate the process variations problems which are mainly based on some form of static approach. However, as the traditional over-design technique becomes impractical, on-chip self-healing which is a dynamic approach has emerged as a promising methodology to address the variability issue. The key idea of self-healing is to actively monitor the post-manufacturing circuit performance metrics and then adaptively adjust a number of tuning knobs, such as bias voltage, in order to meet the given performance specifications. This chapter discusses the self-healing mechanism based analog and radio-frequency ICs.

1.1 Introduction

With the aggressive scaling of nanoscale IC technology, large-scale process variation becomes a critical issue for today's analog and RF ICs [1–5]. As the traditional over-design technique becomes impractical, on-chip self-healing has emerged as a promising methodology to address the variability issue [6–20]. The key idea of self-healing is to actively monitor the post-manufacturing circuit performance metrics and then adaptively adjust a number of tuning knobs (e.g., bias voltage) in order to meet the given performance specifications.

To practically implement on-chip self-healing, a large number of performance metrics must be measured accurately and inexpensively by on-chip sensors. Such a

[1] Carnegie Mellon University, Pittsburgh, PA, USA
[2] IBM T. J. Watson Research Center, Yorktown Heights, NY, USA

measurement task, however, is not trivial, because many analog and RF performance metrics (e.g., phase noise) cannot be easily measured by on-chip sensors. For this reason, alternate test methodology, also called indirect performance sensing, has recently attracted great attention [6, 7, 21], [9–14], [16–20], where the performance of interest (PoI) is not directly measured by an on-chip sensor. Instead, it is accurately predicted from a set of other performance metrics, referred to as the performances of measurement (PoMs) that are highly correlated with PoI and are easy to measure. Toward this goal, indirect sensor modeling is a critical task where the objective is to build a mathematical model to capture the correlation between PoI and PoMs so that PoI can be accurately predicted from PoMs. To achieve this goal, PoMs and PoI are first measured from several training chips, and then indirect sensor models are constructed off-line based on these measurement data. Such indirect sensor models are eventually stored in an on-chip microcontroller for self-healing.

To describe an indirect senor, its model coefficients are stored in an on-chip microcontroller as fixed-point values. A complex model that is composed of many model terms would consume massive hardware resources, since a large number of model coefficients must be stored. Furthermore, during on-chip self-healing, an indirect sensor model is repeatedly evaluated to predict the corresponding PoI based on different PoMs and, therefore, a compact model could dramatically reduce the computational cost. Here, the computational cost accounts for on-chip multiplication and addition, and multiplication dominates the overall computational cost. For these reasons, an indirect sensor model should be compact in order to minimize the cost of on-chip self-healing.

Such a modeling task, however, is nontrivial since there is a tradeoff between the model complexity and the model accuracy. In general, it is likely that an oversimplified model will induce a large modeling error. Here, how to construct a compact indirect sensor model without sacrificing its modeling accuracy remains an open question. In addition, these indirect sensor models must be repeatedly calibrated to accommodate the process shift associated with manufacturing lines. Such a model calibration issue has not been extensively studied yet. Hence, there is a strong need to develop a new methodology to facilitate efficient model calibration with low cost (i.e., requiring few additional measurement data). As such, the overhead of indirect performance sensing and, eventually, the overhead of analog and RF self-healing can be minimized.

To address the aforementioned issues, a novel indirect performance sensing approach is described in this chapter [18]. The method consists of two major steps: (i) pre-silicon indirect sensor modeling and (ii) post-silicon indirect sensor calibration. In the first step, a compact indirect sensor model between PoMs and PoI is constructed based on pre-silicon simulation data by using sparse regression (SR) [22], [23]. SR starts with a complicated model template (e.g., a high-order polynomial) that can accurately capture the correlation between PoMs and PoI. L_1-norm regularization is then applied, resulting in a convex optimization problem which can be efficiently solved to determine the most important model terms in the template without sacrificing any modeling accuracy. Other model coefficients corresponding to the unimportant terms are simply set to zero and are ignored in the final indirect

sensor model. Intuitively, the unimportant model terms have negligible contribution for accurately predicting the value of PoI and, hence, can be discarded to minimize the self-healing cost.

Furthermore, in the second step, an indirect sensor model is repeatedly calibrated based on post-silicon measurement data. To perform efficient model calibration with low cost, a novel Bayesian model fusion (BMF) technique is applied. The key idea of BMF is to combine the old (i.e., before process shift) indirect sensor model with very few new (i.e., after process shift) measurement data to generate a new model that is aligned with the new process condition. Mathematically, the old model is encoded as prior knowledge, and a Bayesian inference is derived to optimally fit the new model by maximum-a-posteriori (MAP) estimation.

Finally, an on-chip self-healing flow is presented where the indirect sensor models are extracted by the described indirect performance sensing technique. Two circuit examples designed in a 32 nm CMOS SOI process are used to validate the aforementioned on-chip self-healing flow. Our experimental results demonstrate that the parametric yield and circuit performance is significantly improved for a wafer after self-healing is applied.

The remainder of this chapter is organized as follows. In Section 1.2, we will present an overview of this novel indirect sensing methodology. The mathematical details for pre-silicon indirect sensor modeling and post-silicon indirect sensor calibration will be described in Sections 1.3 and 1.4, respectively. In Section 1.5, an on-chip self-healing flow based on this novel indirect sensing approach is presented. A 25 GHz differential Colpitts voltage-controlled oscillator (VCO) and a 60 GHz low-noise amplifier (LNA) designed in a 32 nm CMOS SOI process are used to validate this on-chip self-healing flow in Section 1.6. Finally, we conclude in Section 1.7.

1.2 Indirect performance sensing

Without loss of generality, we denote PoI as f and PoMs as:

$$\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_M]^T \tag{1.1}$$

where M stands for the number of performance metrics belonging to PoMs. The objective of indirect performance sensing is to accurately predict the PoI f from the PoMs $\mathbf{x}$ that are highly correlated with f and can be easily measured by on-chip sensors.

Generating an indirect sensor model $f(\mathbf{x})$ consists of two major steps:

- **Pre-silicon indirect sensor modeling** aims to construct a compact model $f(\mathbf{x})$ that can accurately capture the correlation between the PoI f and the PoMs $\mathbf{x}$ based on pre-silicon simulation data.
- **Post-silicon indirect sensor calibration** aims to calibrate the indirect sensor model $f(\mathbf{x})$ based on post-silicon measurement data. Such model calibration must be repeatedly performed in order to accommodate the process shift associated with manufacturing lines.

We start with a generic and complicated model template (e.g., a high-order polynomial) to accurately capture the mapping between PoI and PoMs. The reason we choose a generic model is simply because we do not know the relation between PoI and PoMs in advance. Mathematically, we can write the model $f(\mathbf{x})$ as the linear combination of several basis functions:

$$f(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k \cdot b_k(\mathbf{x}) \tag{1.2}$$

where $\{b_k(\mathbf{x});\ k = 1, 2, \ldots, K\}$ are the basis functions (e.g., linear and quadratic polynomials), $\{\alpha_k; k = 1, 2, \ldots, K\}$ are the model coefficients, and K is the total number of basis functions.

Such a complicated model, though accurate, consumes considerable hardware resources to implement, as all model coefficients must be stored in an on-chip microcontroller to perform on-chip self-healing. To reduce the overhead of on-chip self-healing, we aim to select a small set of basis functions during pre-silicon modeling without surrendering any accuracy. Such a basis function selection task, however, is extremely challenging due to the tradeoff between the model complexity and the modeling error. In general, an over-simplified model is likely to have a large modeling error. SR [22], [23] is applied to efficiently address the aforementioned basis function selection problem. More details about pre-silicon indirect sensor modeling via SR will be discussed in Section 1.3.

Furthermore, at the post-silicon stage, the indirect sensor must be repeatedly calibrated to accommodate the process shift associated with manufacturing lines. Since post-silicon measurement is extremely expensive, sensor calibration must be accomplished with very few post-silicon measurement data to facilitate efficient generation of accurate indirect sensor models and, eventually, minimize the overhead of on-chip self-healing. To this end, a novel BMF technique is applied to keep the calibration cost affordable. The details about post-silicon indirect sensor calibration via BMF will be presented in Section 1.4.

1.3 Pre-silicon indirect sensor modeling via SR

In this section, we aim to construct a compact indirect sensor model to accurately capture the relation between PoI and PoMs. Since the mapping from PoMs to PoI is not known in advance, we start with a generic and complicated model template consisting of a large number of basis functions (e.g., a high-order polynomial), as shown in (1.2). Our objective here is to automatically identify a small set of most important basis functions and then determine their corresponding model coefficients based on pre-silicon simulation data.

To start with, we first collect a number of pre-silicon simulation samples $\{(\mathbf{x}^{(n)}, f^{(n)}); n = 1, \ldots, N\}$, where $\mathbf{x}^{(n)}$ and $f^{(n)}$ denote the values of $\mathbf{x}$ and f for the nth sampling point, respectively, and N denotes the total number of

sampling points. Based on these sampling points, a set of linear equations can be expressed as:

$$\mathbf{B}^T \cdot \boldsymbol{\alpha} = \mathbf{f} \tag{1.3}$$

where

$$\mathbf{B} = \begin{bmatrix} b_1(\mathbf{x}^{(1)}) & b_1(\mathbf{x}^{(2)}) & \cdots & b_1(\mathbf{x}^{(N)}) \\ b_2(\mathbf{x}^{(1)}) & b_2(\mathbf{x}^{(2)}) & \cdots & b_2(\mathbf{x}^{(N)}) \\ \vdots & \vdots & \vdots & \vdots \\ b_K(\mathbf{x}^{(1)}) & b_K(\mathbf{x}^{(2)}) & \cdots & b_K(\mathbf{x}^{(N)}) \end{bmatrix} \tag{1.4}$$

$$\boldsymbol{\alpha} = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_K]^T \tag{1.5}$$

$$\mathbf{f} = [f^{(1)} \quad f^{(2)} \quad \cdots \quad f^{(N)}]^T \tag{1.6}$$

One simple approach to solve the model coefficients $\boldsymbol{\alpha}$ is to apply the traditional ordinary least squares (OLS) fitting method [24]. OLS determines the model coefficients $\boldsymbol{\alpha}$ by solving the following optimization problem:

$$\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad \|\mathbf{B}^T \cdot \boldsymbol{\alpha} - \mathbf{f}\|_2^2 \tag{1.7}$$

where $\| \bullet \|_2$ denotes the L_2-norm of a vector. Intuitively, OLS intends to find a solution $\boldsymbol{\alpha}$ that can minimize the mean squared modeling error.

As mentioned at the beginning of this section, we aim to identify a small set of important basis functions from a large number of possible candidates. All other unimportant basis functions will be discarded due to their negligible contribution for accurately predicting the value of PoI. From this point of view, all model coefficients associated with these unimportant basis functions should be set to zero. Hence, identifying the most important basis functions is equivalent to finding a sparse solution $\boldsymbol{\alpha}$ for the linear equation in (1.3). The OLS formulation in (1.7) poses no constraint on the sparsity of $\boldsymbol{\alpha}$. In other words, the unconstrained optimization in (1.7) used by OLS cannot fit our need of basis function selection. Realizing this limitation of OLS, SR, instead, solves an L_1-norm regularization problem. Before presenting the L_1-norm regularization formulation in Section 1.3.2, we first show the idea of L_0-norm regularization in Section 1.3.1.

1.3.1 L_0-norm regularization

L_0-norm regularization formulates the following optimization to solve the sparse solution for $\boldsymbol{\alpha}$:

$$\begin{aligned} &\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad \|\mathbf{B}^T \cdot \boldsymbol{\alpha} - \mathbf{f}\|_2^2 \\ &\text{subject to} \quad \|\boldsymbol{\alpha}\|_0 \leq \lambda \end{aligned} \tag{1.8}$$

where $\| \bullet \|_0$ stands for the L_0-norm of a vector. The L_0-norm $\|\boldsymbol{\alpha}\|_0$ equals the number of non-zeros in the vector $\boldsymbol{\alpha}$. It measures the sparsity of $\boldsymbol{\alpha}$. Therefore, by directly constraining the L_0-norm, the optimization in (1.8) attempts to find a sparse solution $\boldsymbol{\alpha}$ that minimizes the least squares error.

The parameter λ in (1.8) provides a tradeoff between the sparsity of the solution $\boldsymbol{\alpha}$ and the minimal value of the cost function $\|\mathbf{B}^T \cdot \boldsymbol{\alpha} - \mathbf{f}\|_2^2$. For instance, a large λ is likely to result in a small modeling error, but meanwhile it will increase the number of non-zeros in $\boldsymbol{\alpha}$. It is important to note that if the vector $\boldsymbol{\alpha}$ contains many non-zeros, a large number of model coefficients have to be stored in the on-chip microcontroller to predict the PoI and, hence, the cost of indirect performance sensing can be overly expensive. In practice, the value of λ must be appropriately set to accurately predict the PoI with a small set of basis functions. To find the optimal value of λ, we must accurately estimate the modeling error for different λ values. To avoid over-fitting, we cannot simply measure the modeling error from the set of sampling data that is used to calculate the model coefficients. Instead, modeling error must be measured from an independent data set.

To determine the modeling error for a given λ value, we adopt the idea of Q-fold cross-validation from the statistics community [24]. Namely, we partition the entire data set into Q groups, as shown by the example in Figure 1.1. Modeling error is estimated from Q independent runs. In each run, one of the Q groups is used to estimate the modeling error, and all other groups are used to calculate the model coefficients. Note that the training data for coefficient estimation and the testing data for error estimation are not overlapped. Hence, over-fitting can be easily detected.

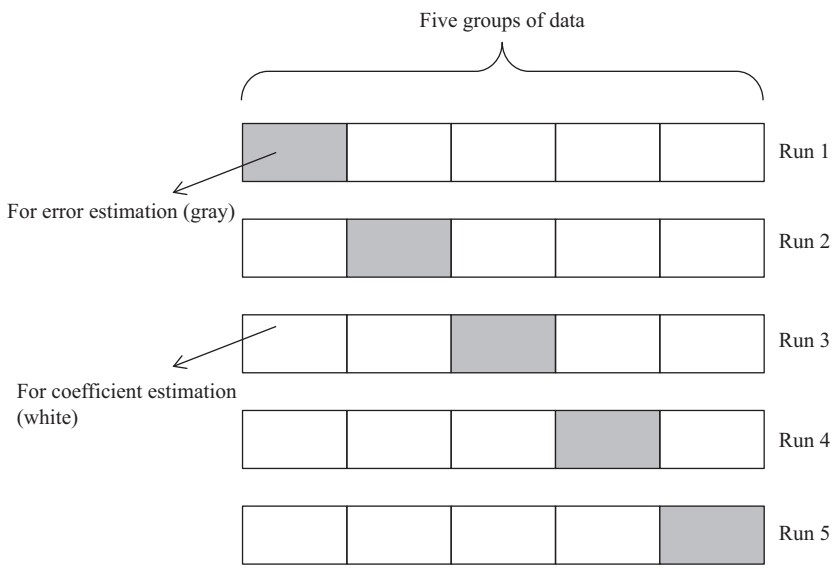


Figure 1.1 A five-fold cross-validation partitions the data set into five groups, and modeling error is estimated from five independent runs

In addition, different groups should be selected for error estimation in different runs. As such, each run results in an error value e_q ($q = 1, 2, \ldots, Q$) that is measured from a unique group of the data set. The final modeling error is computed as the average of e_q ($q = 1, 2, \ldots, Q$), i.e., $e = (e_1 + e_2 + \cdots + e_Q)/Q$. The major drawback of cross-validation is the need to repeatedly extract the model coefficients for Q times. However, for our circuit modeling application, the overall computational cost is dominated by collecting data. Hence, the computational overhead by cross-validation is almost negligible. More details about cross-validation can be found in Reference 24.

While the L_0-norm regularization can effectively guarantee a sparse solution $\boldsymbol{\alpha}$, the optimization in (1.8) is NP hard [23] and, hence, is extremely difficult to solve. A more efficient technique to find sparse solution is based on L_1-norm regularization – a relaxed version of L_0-norm, as described in the next subsection.

1.3.2 L_1-norm regularization

L_1-norm regularization formulates the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \|\mathbf{B}^T \cdot \boldsymbol{\alpha} - \mathbf{f}\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{\alpha}\|_1 \leq \lambda \end{aligned} \tag{1.9}$$

where $\| \bullet \|_1$ denotes the L_1-norm of a vector (i.e., the summation of the absolute values of all elements in the vector):

$$\|\boldsymbol{\alpha}\|_1 = |\alpha_1| + |\alpha_2| + \cdots + |\alpha_K| \tag{1.10}$$

The L_1-norm regularization in (1.9) can be re-formulated as a convex optimization problem. Introduce a set of slack variables $\{\beta_i; i = 0, 1, \ldots, K\}$ and re-write (1.9) into the following equivalent form [25]:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \|\mathbf{B}^T \cdot \boldsymbol{\alpha} - \mathbf{f}\|_2^2 \\ \text{subject to} \quad & \beta_1 + \beta_2 + \cdots + \beta_K \leq \lambda \\ & -\beta_i \leq \alpha_i \leq \beta_i \qquad (i = 1, 2, \ldots, K) \end{aligned} \tag{1.11}$$

In (1.11), the cost function is quadratic and positive semidefinite. Hence, it is convex. All constraints are linear and, therefore, the resulting constraint set is a convex polytope. For these reasons, the L_1-norm regularization in (1.11) is a convex optimization problem, and it can be solved by various efficient and robust algorithms, e.g., the interior-point method [25].

The aforementioned L_1-norm regularization is much more computationally efficient than the L_0-norm regularization that is NP hard. This is the major motivation to replace L_0-norm by L_1-norm. Please note that unlike the conventional OLS that minimizes the mean squared error only, the formulation in (1.9) minimizes the mean squared error subject to an L_1-norm constraint posed on the model coefficients $\boldsymbol{\alpha}$.

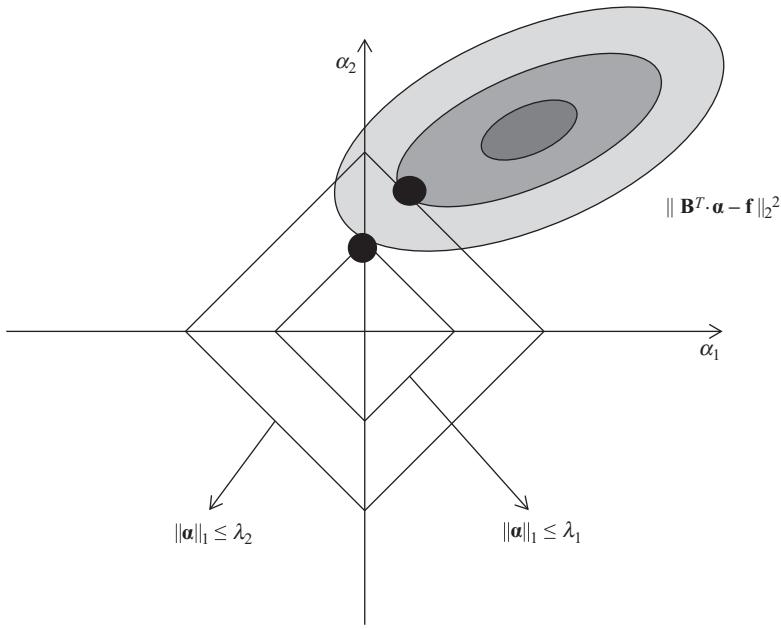


Figure 1.2 L_1-norm regularization $\|\boldsymbol{\alpha}\|_1 \leq \lambda$ results in a sparse solution (i.e., $\alpha_1 = 0$), if λ is sufficiently small (i.e., $\lambda = \lambda_1$)

It, in turn, promotes a sparse solution of $\boldsymbol{\alpha}$ [22], [23] that is desired by our application of basis function selection for on-chip self-healing.

To understand the connection between L_1-norm regularization and sparse solution, we consider a two-dimensional example (i.e., $\boldsymbol{\alpha} = [\alpha_1 \ \alpha_2]^T$), as shown in Figure 1.2. Since the cost function $\|\mathbf{B}^T \cdot \boldsymbol{\alpha} - \mathbf{f}\|_2^2$ is quadratic and positive semidefinite in terms of $\boldsymbol{\alpha}$, its contour lines can be represented by multiple ellipsoids. On the other hand, the constraint $\|\boldsymbol{\alpha}\|_1 \leq \lambda$ corresponds to a number of rotated squares, associated with different values of λ. For example, two of such squares are shown in Figure 1.2 where $\lambda_1 < \lambda_2$.

Studying Figure 1.2, we would notice that if λ is large (e.g., $\lambda = \lambda_2$), both α_1 and α_2 are not zero. However, as λ decreases (e.g., $\lambda = \lambda_1$), the contour of $\|\mathbf{B}^T \cdot \boldsymbol{\alpha} - \mathbf{f}\|_2^2$ eventually intersects the polytope $\|\boldsymbol{\alpha}\|_1 \leq \lambda$ at one of its vertices. It, in turn, implies that one of the coefficients (i.e., α_1 in this case) becomes exactly zero. Hence, by decreasing λ of the L_1-norm regularization in (1.9), we can pose a strong constraint for sparsity and force a sparse solution of $\boldsymbol{\alpha}$ for basis function selection. This intuitively explains why L_1-norm regularization guarantees sparsity, as is the case for L_0-norm regularization.

In addition, various theoretical studies from the statistics community demonstrate that under some general assumptions, both L_1-norm regularization and L_0-norm regularization result in the same solution [26]. Roughly speaking, if the K-dimensional vector $\boldsymbol{\alpha}$ contains W non-zeros and the linear equation $\mathbf{B}^T \cdot \boldsymbol{\alpha} = \mathbf{f}$ is well-conditioned,

the solution $\boldsymbol{\alpha}$ can be uniquely determined by L_1-norm regularization from Z sampling points, where Z is in the order of $O(WlogK)$ [26]. Note that Z (the number of sampling points) is a logarithm function of K (the number of unknown coefficients). It, in turn, provides the theoretical foundation that by solving the sparse solution of an underdetermined equation, a large number of model coefficients can be uniquely determined from a small number of sampling points.

Similar to the L_0-norm regularization shown in (1.8), the parameter λ in (1.9) provides a tradeoff between the sparsity of the solution $\boldsymbol{\alpha}$ and the modeling error. For instance, a large λ is likely to result in a small modeling error, but meanwhile it will increase the number of non-zeros in $\boldsymbol{\alpha}$. To find the optimal value of λ, we can apply the aforementioned Q-fold cross-validation [24].

So far, we only consider how to reduce the number of basis functions (i.e., the number of non-zeros in $\boldsymbol{\alpha}$) in order to save on-chip self-healing cost. Actually, different basis functions may involve different number of multiplications, and the computational cost to calculate each basis function when evaluating the indirect sensor can be quite different. For instance, $\alpha_1 \cdot x$ requires only one multiplication, while $\alpha_2 \cdot x^3$ needs three multiplications. To further reduce the computational cost, we can assign different weights for different coefficients (e.g., a small weight for α_1 while a large weight for α_2) in the constraint of (1.9). Intuitively, a coefficient with a larger weight is more likely to be set to zero in a weighted L_1-norm regularization [25]. Because of the space limitation, the extended version of (1.9) to handle weighted $\boldsymbol{\alpha}$ is not mentioned here.

1.3.3 Accuracy of L_1-norm regularization

Whether the accuracy of the L_1-norm regularization in (1.9) can be quantitatively measured is still an open question. In other words, we need to answer the following two questions:

1. Can the L_1-norm regularization find the exact solution $\boldsymbol{\alpha}$ for the underdetermined linear equation $\mathbf{B}^T \cdot \boldsymbol{\alpha} = \mathbf{f}$?
2. If the answer is yes, what are the sufficient conditions to guarantee the finding of the exact solution $\boldsymbol{\alpha}$?

Next, we will answer these open questions by studying several important statistics theorems. It has been proven in References 26–29 that given the linear equation $\mathbf{B}^T \cdot \boldsymbol{\alpha} = \mathbf{f}$ in (1.3), the accuracy of the L_1-norm regularization depends on the orthonormality of the column vectors of the matrix $\mathbf{B}^T$. To intuitively illustrate this concept, we first consider a trivial case where the number of equations (i.e., N) equals the number of unknowns (i.e., K) and, hence, $\mathbf{B}^T$ is a square matrix. Furthermore, we assume that all column vectors of $\mathbf{B}^T$ are orthonormal, i.e., $\mathbf{B}^T$ is an orthogonal matrix with $\mathbf{B} \cdot \mathbf{B}^T = \mathbf{I}$ where $\mathbf{I}$ is an identity matrix. In this trivial case, the exact solution $\boldsymbol{\alpha}$ of $\mathbf{B}^T \cdot \boldsymbol{\alpha} = \mathbf{f}$ can be accurately determined as:

$$\boldsymbol{\alpha} = \mathbf{B} \cdot \mathbf{f} \tag{1.12}$$

In practice, since collecting samples is very expensive, the linear equation $\mathbf{B}^T \cdot \boldsymbol{\alpha} = \mathbf{f}$ in (1.3) is underdetermined and the matrix $\mathbf{B}^T$ has more columns than rows (i.e., $N < K$). It is impossible for all columns of $\mathbf{B}^T$ to be orthonormal. In this case, it turns out that the solution $\boldsymbol{\alpha}$ can be accurately found if the columns of $\mathbf{B}^T$ are approximately orthonormal. Based on the theorems of compressed sensing [26–29], the "orthonormality" of a matrix $\mathbf{B}^T$ can be quantitatively measured by its restricted isometry property (RIP).

Definition 1.1. *A matrix $\mathbf{B}^T$ satisfies the RIP of order W with constant $\delta_W < 1$, if the inequality:*

$$(1 - \delta_W) \cdot \|\boldsymbol{\alpha}\|_2^2 \leq \|\mathbf{B}^T \cdot \boldsymbol{\alpha}\|_2^2 \leq (1 + \delta_W) \cdot \|\boldsymbol{\alpha}\|_2^2 \tag{1.13}$$

holds for every vector $\boldsymbol{\alpha}$ that contains only W non-zero elements.

If all columns of the matrix $\mathbf{B}^T$ are almost orthonormal, RIP should be satisfied with a large W and a small δ_W. In the extreme case where $\mathbf{B}^T$ is exactly an orthogonal matrix, $\|\mathbf{B}^T \cdot \boldsymbol{\alpha}\|_2$ is equal to $\|\boldsymbol{\alpha}\|_2$ for every vector $\boldsymbol{\alpha}$, since the linear transformation by an orthogonal matrix does not change the L_2-norm of the vector $\boldsymbol{\alpha}$ [30]. Hence, RIP is satisfied with $W = K$ and $\delta_W = 0$.

The concept of RIP has been successfully applied to assess the inherent difficulty of finding the exact solution $\boldsymbol{\alpha}$ from the underdetermined linear equation $\mathbf{B}^T \cdot \boldsymbol{\alpha} = \mathbf{f}$ in (1.3). For example, the following theorem has been shown in Reference 26.

Theorem 1.1. *The L_1-norm regularization in (1.9) guarantees to find the exact solution $\boldsymbol{\alpha}$ of the underdetermined linear equation $\mathbf{B}^T \cdot \boldsymbol{\alpha} = \mathbf{f}$ in (1.3), if the following three conditions are all satisfied:*

1. *The solution vector $\boldsymbol{\alpha}$ contains at most W non-zeros.*
2. *The matrix $\mathbf{B}^T$ satisfies the RIP of order $2W$ with constant $\delta_{2W} < 1$ and the RIP of order $3W$ with constant $\delta_{3W} < 1$.*
3. *The two RIP constants δ_{2W} and δ_{3W} further satisfy the inequality $\delta_{2W} + \delta_{3W} < 1$.*

Note that the conditions in Theorem 1.1 are sufficient but not necessary. A number of other sufficient conditions have also been derived in the literature. More details can be found in Reference 26.

While RIP offers a solid theoretical foundation to assess the accuracy of the L_1-norm regularization, computing the RIP constant δ_W for a given matrix $\mathbf{B}^T$ is an NP-hard problem [26–29]. For this reason, an alternative metric, *coherence*, has been proposed to measure the orthonormality of a matrix $\mathbf{B}^T$ [28].

Definition 1.2. *Given a matrix $\mathbf{B}^T$ for which every column vector has unit length (i.e., unit L_2-norm), its coherence is defined as:*

$$\mu = \max_{i \neq j} |\langle \mathbf{B}_i^T, \mathbf{B}_j^T \rangle| \tag{1.14}$$

where $\mathbf{B}_i^T$ and $\mathbf{B}_j^T$ denote the ith and jth columns of $\mathbf{B}^T$, respectively, and $\langle \bullet, \bullet \rangle$ stands for the inner product of two vectors.

Similar to RIP, the coherence value μ in (1.14) offers a quantitative criterion to judge if the columns of the matrix $\mathbf{B}^T$ are approximately orthonormal. For instance, if all columns of $\mathbf{B}^T$ are orthonormal, the coherence value μ reaches the minimum (i.e., zero); otherwise, the coherence value μ is always greater than zero.

While the RIP constant δ_W in (1.13) is difficult to compute, the coherence value μ in (1.14) can be easily calculated by the inner product of column vectors. Once μ is known, the RIP constant δ_W is bounded by [28]:

$$\delta_W \leq \mu \cdot (W - 1) \tag{1.15}$$

where W denotes the order of RIP. In other words, while the exact value of the RIP constant δ_W is unknown, its upper bound can be efficiently estimated by coherence. This, in turn, offers a computationally tractable way to verify the sufficient conditions in Theorem 1.1. More details on coherence and its applications can be found in Reference 28.

The aforementioned discussions summarize the theoretical framework to justify the accuracy of the L_1-norm regularization. It demonstrates a number of sufficient conditions which guarantee to find the exact sparse solution $\boldsymbol{\alpha}$ from the under-determined linear equation $\mathbf{B}^T \cdot \boldsymbol{\alpha} = \mathbf{f}$. In our application, the number of non-zeros in the vector $\boldsymbol{\alpha}$ is not known in advance. Hence, it can be difficult to verify the conditions in Theorem 1.1 and then determine if the exact solution $\boldsymbol{\alpha}$ is accurately solved. However, the theoretical results summarized here demonstrate the importance of column orthonormality for the matrix $\mathbf{B}^T$ in (1.3). Due to the page limit, how to improve the column orthonormality and, hence, enhance the accuracy of SR is not discussed here.

The aforementioned SR method based on L_1-norm regularization can be efficiently applied to pre-silicon basis function selection and model coefficient estimation. However, the device models used for pre-silicon simulation are not perfectly accurate and may differ from the post-silicon measurement results. For this reason, there is a strong need to further calibrate the proposed indirect sensor models based on post-silicon measurement data, as will be discussed in the next section.

1.4 Post-silicon indirect sensor calibration via Bayesian model fusion

The objective of post-silicon indirect sensor calibration is to further correct the modeling error posed by pre-silicon simulation and also accommodate the process shift associated with manufacturing lines. One straightforward approach for sensor calibration is to collect a large amount of post-silicon measurement data and then completely re-fit the indirect sensor model. Such a simple approach, however, can be practically unaffordable, since post-silicon testing is time-consuming and, hence, it is overly expensive to collect a large set of post-silicon measurement data.

To address this cost issue, we apply a novel statistical framework, referred to as Bayesian model fusion (BMF) [31], for efficient post-silicon sensor calibration.

BMF relies on an important observation that even though the simulation and/or measurement data collected at multiple stages (e.g., pre-silicon vs. post-silicon) are not exactly identical, they are expected to be strongly correlated. Hence, it is possible to borrow the data from an early stage (e.g., pre-silicon) for sensor calibration at a late stage (e.g., post-silicon). As such, only few post-silicon data should be measured at the late stage and, hence, the cost of sensor calibration is substantially reduced.

More specifically, our indirect sensor models are initially fitted by using the early-stage (e.g., pre-silicon) data. Next, the early-stage sensor model is encoded as our prior knowledge. Finally, the indirect sensor model is further calibrated by applying Bayesian inference with very few late-stage (e.g., post-silicon) measurement data. Here, by "fusing" the early-stage and late-stage sensor models through Bayesian inference, the amount of required measurement data (hence, the measurement cost) can be substantially reduced at the late stage.

To fully understand the BMF method, let us consider two different models: the early-stage model $f_E(\mathbf{x})$ and the late-stage model $f_L(\mathbf{x})$:

$$f_E(\mathbf{x}) = \sum_{k=1}^{K} \alpha_{E,k} \cdot b_k(\mathbf{x}) + \varepsilon_E \tag{1.16}$$

$$f_L(\mathbf{x}) = \sum_{k=1}^{K} \alpha_{L,k} \cdot b_k(\mathbf{x}) + \varepsilon_L \tag{1.17}$$

where $\{b_k(\mathbf{x}); k = 1, 2, \ldots, K\}$ are the basis functions selected by SR at the early stage, $\{\alpha_{E,k}; k = 1, 2, \ldots, K\}$ and $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ contain the early-stage and late-stage model coefficients, respectively, and ε_E and ε_L denote the modeling error associated with the early-stage and late-stage models, respectively.

The early-stage model $f_E(\mathbf{x})$ in (1.16) is fitted by using the early-stage (e.g., pre-silicon) data. Hence, we assume that the early-stage model coefficients $\{\alpha_{E,k}; k = 1, 2, \ldots, K\}$ are already known, before fitting the late-stage model $f_L(\mathbf{x})$ in (1.17) based on the late-stage (e.g., post-silicon) measurement data. The objective of BMF is to accurately determine the late-stage model coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ by combining the early-stage model coefficients $\{\alpha_{E,k}; k = 1, 2, \ldots, K\}$ with very few late-stage measurement data.

BMF method consists of two major steps: (i) statistically extracting the prior knowledge from the early-stage model coefficients $\{\alpha_{E,k}; k = 1, 2, \ldots, K\}$ and encoding it as a prior distribution and (ii) optimally determining the late-stage model coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ by MAP estimation. In what follows, we will describe these two steps in detail.

1.4.1 Prior knowledge definition

Since the two models $f_E(\mathbf{x})$ and $f_L(\mathbf{x})$ in (1.16) and (1.17) both approximate the mathematical mapping from PoMs to PoI, we expect that the model coefficients

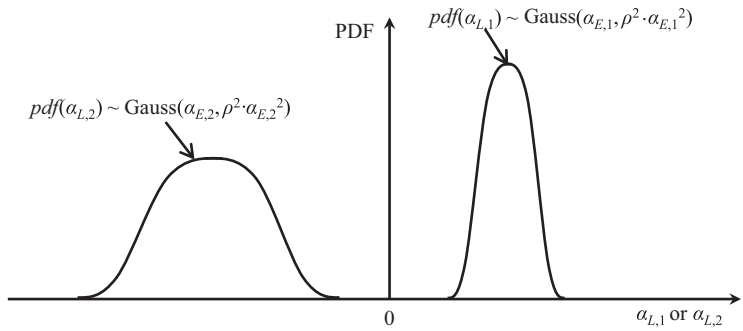


Figure 1.3 A simple example of the prior distribution is shown for two model coefficients $\alpha_{L,1}$ and $\alpha_{L,2}$. The coefficient $\alpha_{L,1}$ possibly takes a small magnitude, since its prior distribution is narrowly peaked around a small value. The coefficient $\alpha_{L,2}$ possibly takes a large magnitude, since its prior distribution widely spreads over a large value

$\{\alpha_{E,k}; k = 1, 2, \ldots, K\}$ and $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ are similar. On the other hand, $f_E(\mathbf{x})$ and $f_L(\mathbf{x})$ cannot be exactly identical, since they represent the indirect sensor models at two different stages. To statistically encode the "common" information between $f_E(\mathbf{x})$ and $f_L(\mathbf{x})$, we define a Gaussian distribution as our prior distribution for each late-stage model coefficient $\alpha_{L,k}$:

$$pdf(\alpha_{L,k}) = \frac{1}{\sqrt{2\pi} \cdot \rho \cdot |\alpha_{E,k}|} \cdot \exp\left[-\frac{(\alpha_{L,k} - \alpha_{E,k})^2}{2 \cdot \rho^2 \cdot \alpha_{E,k}^2}\right] \quad k = 1, 2, \ldots, K \tag{1.18}$$

where $\alpha_{E,k}$ and $\rho^2 \cdot \alpha_{E,k}^2$ are the mean and variance of the Gaussian distribution, respectively, and $\rho > 0$ is a parameter that can be determined by cross-validation [24]. Figure 1.3 shows a simple example of the prior distribution for two model coefficients $\alpha_{L,1}$ and $\alpha_{L,2}$ where $|\alpha_{E,1}|$ is small and $|\alpha_{E,2}|$ is large.

The prior distribution in (1.18) has a two-fold meaning. First, the Gaussian distribution $pdf(\alpha_{L,k})$ is peaked at its mean value $\alpha_{E,k}$, implying that the early-stage model coefficient $\alpha_{E,k}$ and the late-stage model coefficient $\alpha_{L,k}$ are likely to be similar. In other words, since the Gaussian distribution $pdf(\alpha_{L,k})$ exponentially decays with $(\alpha_{L,k} - \alpha_{E,k})^2$, it is unlikely to observe a late-stage coefficient $\alpha_{L,k}$ that is extremely different from the early-stage coefficient $\alpha_{E,k}$. Second, the standard deviation of the prior distribution $pdf(\alpha_{L,k})$ is proportional to $|\alpha_{E,k}|$. It means that the absolute difference between the late-stage coefficient $\alpha_{L,k}$ and the early-stage coefficient $\alpha_{E,k}$ can be large (or small), if the magnitude of the early-stage coefficient $|\alpha_{E,k}|$ is large (or small). Restating in words, each late-stage coefficient $\alpha_{L,k}$ has been provided with a relatively equal opportunity to deviate from the corresponding early-stage coefficient $\alpha_{E,k}$.

To complete the definition of the prior distribution for all late-stage model coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$, we further assume that these coefficients are statistically independent, and their joint distribution is represented as:

$$pdf(\boldsymbol{\alpha}_L) = \prod_{k=1}^{K} pdf(\alpha_{L,k}) \tag{1.19}$$

where

$$\boldsymbol{\alpha}_L = [\alpha_{L,1} \quad \alpha_{L,2} \quad \cdots \quad \alpha_{L,K}]^T \tag{1.20}$$

is a vector containing all late-stage coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$. Combining (1.18) and (1.19) yields

$$pdf(\boldsymbol{\alpha}_L) = \frac{1}{(\sqrt{2\pi} \cdot \rho)^K \cdot \prod_{k=1}^{K} |\alpha_{E,k}|} \cdot \exp\left[-\frac{(\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E)^T \cdot \mathbf{A} \cdot (\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E)}{2 \cdot \rho^2}\right] \tag{1.21}$$

where

$$\boldsymbol{\alpha}_E = [\alpha_{E,1} \quad \alpha_{E,2} \quad \cdots \quad \alpha_{E,K}]^T \tag{1.22}$$

is a vector containing all early-stage coefficients $\{\alpha_{E,k}; k = 1, 2, \ldots, K\}$, and

$$\mathbf{A} = \begin{bmatrix} \frac{1}{\alpha_{E,1}^2} & & & \\ & \frac{1}{\alpha_{E,2}^2} & & \\ & & \ddots & \\ & & & \frac{1}{\alpha_{E,K}^2} \end{bmatrix} \tag{1.23}$$

The independence assumption in (1.19) simply implies that we do not know the correlation information among these coefficients as our prior knowledge. The correlation information will be learned from the late-stage measurement data, when the posterior distribution is calculated by MAP estimation in the next sub-section.

1.4.2 MAP estimation

Once the prior distribution is defined, we collect a few (i.e., N) late-stage measurement data $\{(\mathbf{x}^{(n)}, f_L^{(n)}); n = 1, \ldots, N\}$, where $\mathbf{x}^{(n)}$ and $f_L^{(n)}$ are the values of $\mathbf{x}$ and $f_L(\mathbf{x})$ for the nth data point, respectively. These new measurement data can tell us additional information about the difference between early and late stages and, hence, help us to determine the late-stage coefficients $\boldsymbol{\alpha}_L$.

Based on Bayes' theorem [24], the uncertainties of the late-stage coefficients $\boldsymbol{\alpha}_L$ after knowing the data $\{(\mathbf{x}^{(n)}, f_L^{(n)}); n = 1, \ldots, N\}$ can be mathematically described by the following posterior distribution:

$$pdf(\boldsymbol{\alpha}_L|\mathbf{X}, \mathbf{f}_L) \propto pdf(\boldsymbol{\alpha}_L) \cdot pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L) \tag{1.24}$$

where

$$\mathbf{f}_L = [f_L^{(1)} \quad f_L^{(2)} \quad \cdots \quad f_L^{(N)}]^T \tag{1.25}$$

$$\mathbf{X} = [\mathbf{x}^{(1)} \quad \mathbf{x}^{(2)} \quad \cdots \quad \mathbf{x}^{(N)}]^T \tag{1.26}$$

In (1.24), the prior distribution $pdf(\boldsymbol{\alpha}_L)$ is defined by (1.21). The conditional distribution $pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L)$ is referred to as the likelihood function. It measures the probability of observing the new data $\{(\mathbf{x}^{(n)}, f_L^{(n)}); n = 1, \ldots, N\}$.

To derive the likelihood function $pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L)$, we assume that the modeling error ε_L in (1.17) can be represented as a random variable with zero-mean Gaussian distribution:

$$pdf(\varepsilon_L) = \frac{1}{\sqrt{2\pi} \cdot \sigma_0} \cdot \exp\left[-\frac{\varepsilon_L^2}{2 \cdot \sigma_0^2}\right] \sim N(0, \sigma_0^2) \tag{1.27}$$

where the standard deviation σ_0 indicates the magnitude of the modeling error. Similar to the parameter ρ in (1.18), the value of σ_0 can be determined by cross-validation [24]. Since the modeling error associated with the nth data point $(\mathbf{x}^{(n)}, f_L^{(n)})$ is simply one sampling point of the random variable ε_L, it follows the Gaussian distribution:

$$f_L^{(n)} - \sum_{k=1}^{K} \alpha_{L,k} \cdot b_k(\mathbf{x}^{(n)}) \sim N(0, \sigma_0^2) \tag{1.28}$$

Therefore, the probability of observing the nth data point $(\mathbf{x}^{(n)}, f_L^{(n)})$ is

$$pdf[\mathbf{x}^{(n)}, f_L^{(n)}|\boldsymbol{\alpha}_L] = \frac{1}{\sqrt{2\pi} \cdot \sigma_0} \cdot \exp\left\{-\frac{1}{2 \cdot \sigma_0^2} \cdot \left[f_L^{(n)} - \sum_{k=1}^{K} \alpha_{L,k} \cdot b_k(\mathbf{x}^{(n)})\right]^2\right\} \tag{1.29}$$

Note that the likelihood function $pdf[\mathbf{x}^{(n)}, f_L^{(n)}|\boldsymbol{\alpha}_L]$ in (1.29) depends on the late-stage model coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$. Assuming that all data points

$\{(\mathbf{x}^{(n)}, f_L^{(n)}); n = 1, \ldots, N\}$ are independently generated, we can write the likelihood function $pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L)$ as:

$$\begin{aligned} pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L) &= \prod_{n=1}^{N} pdf[\mathbf{x}^{(n)}, f_L^{(n)}|\boldsymbol{\alpha}_L] \\ &= \frac{1}{(\sqrt{2\pi}\cdot\sigma_0)^N} \cdot \exp\left\{-\frac{1}{2\cdot\sigma_0^2}\cdot\sum_{n=1}^{N}[f_L^{(n)} - \sum_{k=1}^{K}\alpha_{L,k}\cdot b_k(\mathbf{x}^{(n)})]^2\right\} \end{aligned} \tag{1.30}$$

Equation (1.30) can be re-written as:

$$pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L) = \frac{1}{(\sqrt{2\pi}\cdot\sigma_0)^N} \cdot \exp\left\{-\frac{(\mathbf{B}^T\cdot\boldsymbol{\alpha}_L - \mathbf{f}_L)^T\cdot(\mathbf{B}^T\cdot\boldsymbol{\alpha}_L - \mathbf{f}_L)}{2\cdot\sigma_0^2}\right\} \tag{1.31}$$

where $\mathbf{B}$, $\boldsymbol{\alpha}_L$ and $\mathbf{f}_L$ are defined in (1.4), (1.20), and (1.25), respectively.

After the new data $\{(\mathbf{x}^{(n)}, f_L^{(n)}); n = 1, \ldots, N\}$ are available, the late-stage coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ can be described by the probability density function $pdf(\boldsymbol{\alpha}_L|\mathbf{X}, \mathbf{f}_L)$ (i.e., the posterior distribution) in (1.24). Depending on the shape of the posterior distribution $pdf(\boldsymbol{\alpha}_L|\mathbf{X}, \mathbf{f}_L)$, the late-stage coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ do not take all possible values with equal probability. If the posterior distribution $pdf(\boldsymbol{\alpha}_L|\mathbf{X}, \mathbf{f}_L)$ reaches its maximum value at $\{\alpha_{L,k}^*; k = 1, 2, \ldots, K\}$, these values $\{\alpha_{L,k}^*; k = 1, 2, \ldots, K\}$ are the optimal estimation of the late-stage coefficients, since these coefficient values are most likely to occur. Such a method is referred to as the MAP estimation in the literature [24].

The aforementioned MAP estimation can be formulated as an optimization problem:

$$\underset{\boldsymbol{\alpha}_L}{\text{maximize}} \quad pdf(\boldsymbol{\alpha}_L|\mathbf{X}, \mathbf{f}_L) \tag{1.32}$$

Substituting (1.24) into (1.32) yields

$$\underset{\boldsymbol{\alpha}_L}{\text{maximize}} \quad pdf(\boldsymbol{\alpha}_L)\cdot pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L) \tag{1.33}$$

Combining (1.21), (1.31), and (1.33), we have

$$\underset{\boldsymbol{\alpha}_L}{\text{maximize}} \ \exp\left[-\frac{(\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E)^T\cdot\mathbf{A}\cdot(\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E)}{2\cdot\rho^2} - \frac{(\mathbf{B}^T\cdot\boldsymbol{\alpha}_L - \mathbf{f}_L)^T\cdot(\mathbf{B}^T\cdot\boldsymbol{\alpha}_L - \mathbf{f}_L)}{2\cdot\sigma_0^2}\right] \tag{1.34}$$

Since the exponential function is monotonically increasing, (1.34) can be re-written as:

$$\underset{\boldsymbol{\alpha}_L}{\text{minimize}} \ \eta\cdot(\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E)^T\cdot\mathbf{A}\cdot(\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E) + (\mathbf{B}^T\cdot\boldsymbol{\alpha}_L - \mathbf{f}_L)^T\cdot(\mathbf{B}^T\cdot\boldsymbol{\alpha}_L - \mathbf{f}_L) \tag{1.35}$$

where

$$\eta = \frac{\sigma_0^2}{\rho^2} \tag{1.36}$$

It is straightforward to prove that the cost function in (1.35) is convex [25]. Hence, its global optimum can be directly solved by applying the first-order optimality condition [25]:

$$\begin{aligned}&\frac{\partial[\eta \cdot (\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E)^T \cdot \mathbf{A} \cdot (\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E) + (\mathbf{B}^T \cdot \boldsymbol{\alpha}_L - \mathbf{f}_L)^T \cdot (\mathbf{B}^T \cdot \boldsymbol{\alpha}_L - \mathbf{f}_L)]}{\partial \boldsymbol{\alpha}_L} \\ &= 2 \cdot \eta \cdot \mathbf{A} \cdot (\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E) + 2 \cdot \mathbf{B} \cdot (\mathbf{B}^T \cdot \boldsymbol{\alpha}_L - \mathbf{f}_L) \\ &= \mathbf{0}\end{aligned} \tag{1.37}$$

Solving the linear equation in (1.37) results in the optimal value of $\boldsymbol{\alpha}_L$:

$$\boldsymbol{\alpha}_L = (\eta \cdot \mathbf{A} + \mathbf{B} \cdot \mathbf{B}^T)^{-1} \cdot (\eta \cdot \mathbf{A} \cdot \boldsymbol{\alpha}_E + \mathbf{B} \cdot \mathbf{f}_L) \tag{1.38}$$

Studying (1.38), we observe that only the value of η is required to find the late-stage model coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ and, hence, we only need to determine η, instead of the individual parameters σ_0 and ρ. In our work, the optimal value of η is determined by cross-validation [24].

Once the optimal η value is found, the late-stage model coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ are calculated from (1.38), and then an updated indirect sensor model $f_L(\mathbf{x})$ in (1.17) is generated to match the late-stage measurement data. Such a calibrated indirect sensor model is eventually stored in an on-chip microcontroller to facilitate efficient on-chip self-healing, as will be discussed in detail in the next section.

Finally, it is important to mention that the post-silicon indirect sensor calibration is performed off-chip and, hence, no hardware overhead is introduced. To further reduce the indirect sensor calibration cost (i.e., with very few number of post-silicon measurement data), we can calibrate the indirect sensor if and only if the new measurement data are not consistent with the old indirect sensor model. In practice, such inconsistency can be detected by measuring a small number of dies from each wafer or lot to estimate the indirect sensing error. If the error is not sufficiently small, the indirect sensor model must be calibrated.

1.5 On-chip self-healing flow

In this section, we will further describe a practical on-chip self-healing flow based on the aforementioned indirect sensing approach. As mentioned earlier, the key idea of on-chip self-healing is to actively monitor the post-manufacturing circuit performance metrics and then adaptively adjust a number of tuning knobs (e.g., bias voltage)

in order to meet the given performance specifications. In this work, we mathematically formulate the self-healing problem as a constrained optimization where one particular performance metric is minimized subject to a set of given performance constraints:

$$\begin{aligned} &\underset{\mathbf{t}}{\text{minimize}} \quad f(\mathbf{t}) \\ &\text{subject to} \quad g_p(\mathbf{t}) \geq s_p \quad (p = 1, 2, \ldots, P) \end{aligned} \tag{1.39}$$

where $\mathbf{t}$ denotes the set of tuning knobs, $f(\mathbf{t})$ denotes the performance metric that we aim to minimize, and $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ denote the other P performance metrics with the given specifications $\{s_p; p = 1, 2, \ldots, P\}$. Take mixer as an example. We aim to minimize the mixer power while keeping its gain and 1 dB compression point larger than their specifications. In this case, $f(\mathbf{t})$ is the mixer power, $g_1(\mathbf{t})$ is the mixer gain, and $g_2(\mathbf{t})$ is the 1 dB compression point.

There are two important clarifications we need to make for the optimization formulation in (1.39). First, the formulation in (1.39) is set up for a circuit where one performance metric $f(\mathbf{t})$ should be minimized while constraining all other performance metrics $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ to their lower bounds $\{s_p; p = 1, 2, \ldots, P\}$. For a circuit where a performance metric $f(\mathbf{t})$ should be maximized, the objective function in (1.39) can be simply modified to $-f(\mathbf{t})$. Similarly, for a circuit where the performance metrics $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ should be constrained to their upper bounds $\{s_p; p = 1, 2, \ldots, P\}$, the constraints in (1.39) can be adjusted as $-g_p(\mathbf{t}) \geq -s_p$ $(p = 1, 2, \ldots, P)$. Second, not all the performance metrics $f(\mathbf{t})$ and $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ in our self-healing circuit can be directly measured by on-chip sensors. For the performance metrics that cannot be easily measured by on-chip sensors, the proposed indirect performance sensing technique is applied to efficiently and accurately predict their values.

To find the optimal solution $\mathbf{t}^*$ in (1.39), an on-chip self-healing flow shown in Figure 1.4 is applied, where the indirect sensors are modeled and calibrated by the SR and BMF techniques described in previous sections. The indirect sensor models are stored and evaluated by a microcontroller for on-chip self-healing. The search algorithm starts with an initial guess $\mathbf{t}_0$. We set $\mathbf{t} = \mathbf{t}_0$, and all performance metrics $f(\mathbf{t})$ and $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ are measured either directly or indirectly. Here, we use the symbol PMs to represent the performance metrics that are directly measured by on-chip sensors, and the symbol PoIs to represent the performance metrics that are estimated by the proposed indirect sensors. In particular, to estimate the PoIs, the corresponding PoMs are first measured by on-chip sensors. Next, the indirect sensor models stored in the on-chip microcontroller are evaluated to predict the PoIs, as shown in Figure 1.4. Based on the performance values $\{f(\mathbf{t}), g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$, $\mathbf{t}$ is updated and the aforementioned process is repeated until the optimal solution $\mathbf{t}^*$ is found. Algorithm 1 summarizes the details of such an optimization flow with indirect performance sensing for on-chip self-healing. Once the optimal solution $\mathbf{t}^*$ is found, tuning knobs are adjusted to the values of $\mathbf{t}^*$ and the self-healing process is complete.

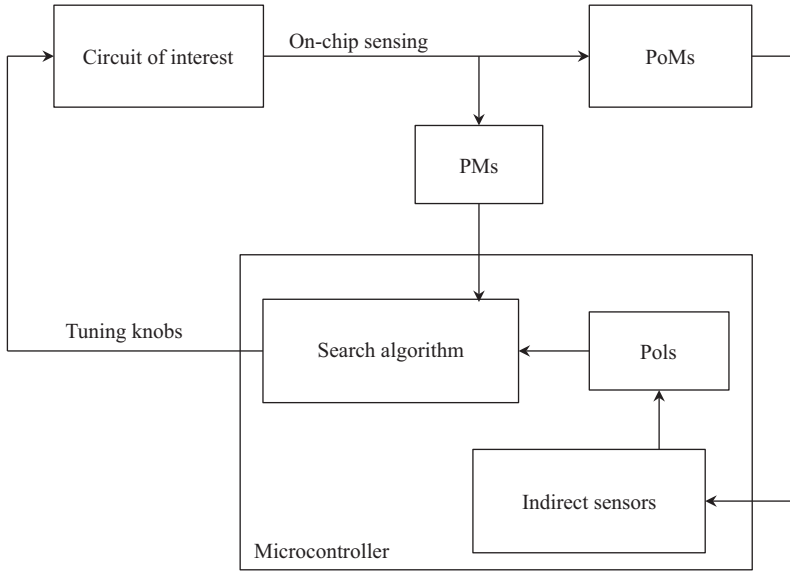


Figure 1.4 A simplified block diagram describes the on-chip self-healing flow

Algorithm 1 On-chip self-healing flow

1. Start with the constrained optimization problem in (1.39) and an initial guess $\mathbf{t}_0$.
2. Set $\mathbf{t} = \mathbf{t}_0$.
3. Measure $f(\mathbf{t})$ and $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ either directly or indirectly.
4. Based on the performance values $\{f(\mathbf{t}), g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$, update $\mathbf{t}$.
5. If $\mathbf{t}$ is the optimal solution, stop iteration. Otherwise, go to Step 3.

For different circuits of interest with different performance metrics and tuning knobs, the search strategy of updating $\mathbf{t}$ in Step 1.4 of Algorithm 1 could be substantially different. For instance, if there is only a small number of (e.g., one or two) tuning knobs, we can apply a simple brute-force search algorithm to find the optimal solution of (1.39). Without loss of generality, we assume that the tuning knobs can take H possible values $\{\mathbf{t}_h; h = 1, 2, \ldots, H\}$. The initial value of $\mathbf{t}$ is set as $\mathbf{t}_1$ in the first iteration, and $\{f(\mathbf{t}_1), g_p(\mathbf{t}_1); p = 1, 2, \ldots, P\}$ are either directly or indirectly measured. Next, in the second iteration, $\mathbf{t}$ is updated to $\mathbf{t}_2$, and $\{f(\mathbf{t}_2), g_p(\mathbf{t}_2); p = 1, 2, \ldots, P\}$ are measured. Similarly, in the hth iteration, $\{f(\mathbf{t}_h), g_p(\mathbf{t}_h); p = 1, 2, \ldots, P\}$ are measured. In the end, we have a large data set $\{f(\mathbf{t}_h), g_p(\mathbf{t}_h); p = 1, 2, \ldots, P,\ h = 1, 2, \ldots, H\}$. The optimal solution $\mathbf{t}^*$ for the optimization in (1.39) can be eventually determined based on the performance values $\{f(\mathbf{t}_h), g_p(\mathbf{t}_h); p = 1, 2, \ldots, P,\ h = 1, 2, \ldots, H\}$. Algorithm 2 summarizes the details of the aforementioned brute-force search algorithm.

Algorithm 2 Brute-force search for on-chip self-healing

1. Start with the constrained optimization problem in (1.39) and H possible values $\{\mathbf{t}_h; h = 1, 2, \ldots, H\}$ for the tuning knobs. Set $h = 1$.
2. Set the tuning knobs to $\mathbf{t} = \mathbf{t}_h$.
3. Measure $f(\mathbf{t})$ and $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ and set $f(\mathbf{t}_h) = f(\mathbf{t})$, and $\{g_p(\mathbf{t}_h) = g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$.
4. If $h < H$, $h = h + 1$ and go to Step 2. Otherwise, go to Step 5.
5. Based on the performance values $\{f(\mathbf{t}_h), g_p(\mathbf{t}_h); p = 1, 2, \ldots, P,\ h = 1, 2, \ldots, H\}$, determine the optimal solution $\mathbf{t}^*$ for the optimization in (1.39).

The brute-force search algorithm (i.e., Algorithm 2), though simple to implement, has no practical utility if we have a large number of tuning knobs. To understand the reason, let us consider the general case of U tuning knobs where the uth tuning knob can take V_u possible values. In this case, we have $H = V_1 \cdot V_2 \cdots V_U$ possible values for $\mathbf{t}$ in (1.39). With the increasing number of tuning knobs, the total number of possible values for these tuning knobs (i.e., H) will dramatically increase, thereby making the brute-force search algorithm quickly intractable. In these cases, other efficient search algorithms (e.g., interior-point method [25]) must be applied to solve the optimization in (1.39) for on-chip self-healing.

Before ending this section, it is important to discuss the design overhead of on-chip self-healing that requires a number of additional circuitries (e.g., on-chip sensors, on-chip microcontroller, etc.), as shown in Figure 1.4. There are several important clarifications we need to make here. First, many analog and RF circuit blocks on the same chip may require self-healing, and they can possibly share the same on-chip sensors and microcontroller. Second, for a typical system-on-chip (SoC) application, the microcontroller is needed for other computing tasks during the normal operation. In other words, the microcontroller is not added for on-chip self-healing only. For these reasons, the design overhead of on-chip self-healing is fairly small, or even negligible, in many application scenarios.

1.6 Case study

1.6.1 25 GHz differential Colpitts VCO

In this subsection, a 25 GHz differential Colpitts VCO designed in a 32 nm CMOS SOI process is used to validate the proposed on-chip self-healing flow based on off-line data analysis. Figure 1.5 shows the simplified schematic of the VCO. It consists of a cross-coupled differential pair connected to two common-gate Colpitts oscillators. The capacitor at the output is tunable so that the VCO frequency can be centered at different frequency bands. The bias voltage V_b is controlled by a digital-to-analog converter (DAC) for self-healing. More details about the VCO design can be found in Reference 32.

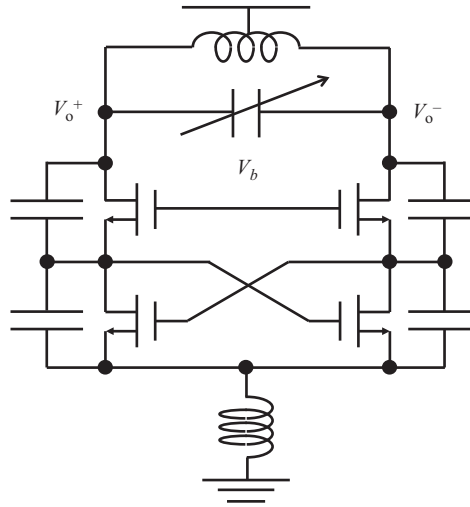


Figure 1.5 Simplified circuit schematic is shown for a Colpitts VCO

Table 1.1 Frequencies and corresponding phase noise specifications

Frequency (GHz)	26.2	24.6	23.3	22.2
PN specification (dBc/Hz)	−123.5	−123.5	−123.5	−123.5

For the VCO shown in Figure 1.5, since we only have one tuning knob (i.e., the bias voltage V_b), the simple brute-force search algorithm described in Algorithm 2 is applied for self-healing. In this example, phase noise is an important PoI, and its specifications derived from the system requirement for four different center frequencies are shown in Table 1.1. If the phase noise value of a VCO is smaller than the given specification at all four frequencies shown in Table 1.1, this VCO is considered as "PASS". Otherwise, we consider it as "FAIL". The objective of self-healing is to find the optimal bias voltage to minimize the phase noise.

Accurately measuring the phase noise at 25 GHz is not trivial. Hence, an indirect sensor is used for on-chip phase noise measurement (i.e., phase noise is considered as a PoI in Figure 1.4). According to Leeson's model [33], oscillation frequency (x_1), oscillation amplitude (x_2), and bias current (x_3), all of which are easy to measure using fully integrated sub-circuits, have strong correlation with phase noise and, hence, are first chosen as PoMs. The sensors used to measure these three PoMs are listed in Table 1.2. An on-chip current sensor to measure bias current will be integrated in our future work. More details about how to measure these three PoMs can be found in Reference 16. In addition, the tuning knob V_b (x_4) is considered as another PoM. Since V_b is directly controlled by a DAC, the digitized value of V_b is known, and no measurement is required. In total, four PoMs (i.e., $\mathbf{x} = [x_1\ x_2\ x_3\ x_4]^T$) are chosen as the indirect sensor inputs and are summarized in Table 1.3. Next, a

Table 1.2 PoMs and measurement sensors

PoM	x_1	x_2	x_3
Measurement sensor	On-chip counter	On-chip peak detector + On-chip 6 bit ADC	Off-chip current sensor

Table 1.3 PoI and PoMs of indirect phase noise sensor

	Performance metric
PoI	Phase noise
PoMs	Oscillation frequency (x_1) Oscillation amplitude (x_2) Bias current (x_3) Bias voltage (x_4)

Table 1.4 Basis functions selected for indirect phase noise sensor

Index	**Term**	**Index**	**Term**	**Index**	**Term**
1	x_2	4	x_1^2	7	$x_3 \cdot x_4$
2	x_3	5	$x_1 \cdot x_2$	8	x_4^2
3	x_4	6	$x_1 \cdot x_4$	9	const

quadratic model template with four input variables (i.e., x_1, x_2, x_3, and x_4) and, hence, 15 polynomial terms in total is used to build the indirect sensor for phase noise. With all 15 polynomial terms, the average modeling error is 0.36 dBc/Hz. SR is then applied to simplify the quadratic model template. Nine polynomial terms are eventually selected by SR, as summarized in Table 1.4. The average modeling error of the simplified quadratic model is 0.41 dBc/Hz. The degradation of the modeling accuracy is negligible (0.05 dBc/Hz only). The accuracy of the simplified quadratic model with nine polynomial terms can be further demonstrated by the scatter plot between the actual phase noise and the predicted phase noise shown in Figure 1.6.

Next, we collect four PoMs and phase noise at all possible bias voltages from a silicon wafer that contains 61 functional VCOs. The VCOs that are not functioning in this wafer are not considered here. These data are further used to calibrate the indirect phase noise sensor to improve its accuracy. Without self-healing, the parametric yield achieved by using a fixed bias voltage for all the VCOs on the wafer is summarized in Table 1.5. Here, bias code denotes a digitized bias voltage, and parametric yield is defined as the ratio between the number of functional VCOs that can meet all four given phase noise specifications shown in Table 1.1 and the total number of functional VCOs. From Table 1.5, we can see that the best parametric yield achieved

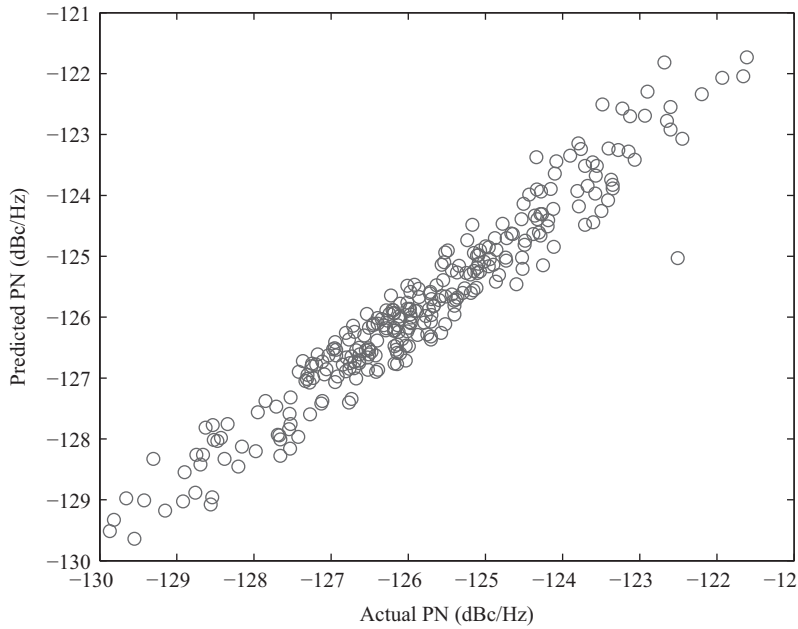


Figure 1.6 Scatter plot is shown for the actual phase noise and the predicted phase noise based on the simplified quadratic model

Table 1.5 Parametric yield of the wafer by using a fixed bias voltage

Bias code	**1**	**2**	**3**	**4**	**5**	**6**
Parametric yield	0	0	1.64%	11.48%	3.28%	0

by using a fixed bias voltage (i.e., bias code is 4) is only 11.48%. If other bias voltages are selected during the design, the parametric yield is even worse, which is almost zero for this wafer. It, in turn, serves as an excellent design case to demonstrate the importance of self-healing. For testing and comparison purposes, three different self-healing methods are implemented:

- **Ideal:** The optimal bias voltage is determined by directly measuring the phase noise with an off-chip tester for all bias voltages. As a result, no indirect phase noise sensor is needed, and all the off-chip measurement data from the wafer will be used. This approach is not considered as on-chip self-healing; however, it provides the upper bound of the yield improvement that can be achieved by self-healing.

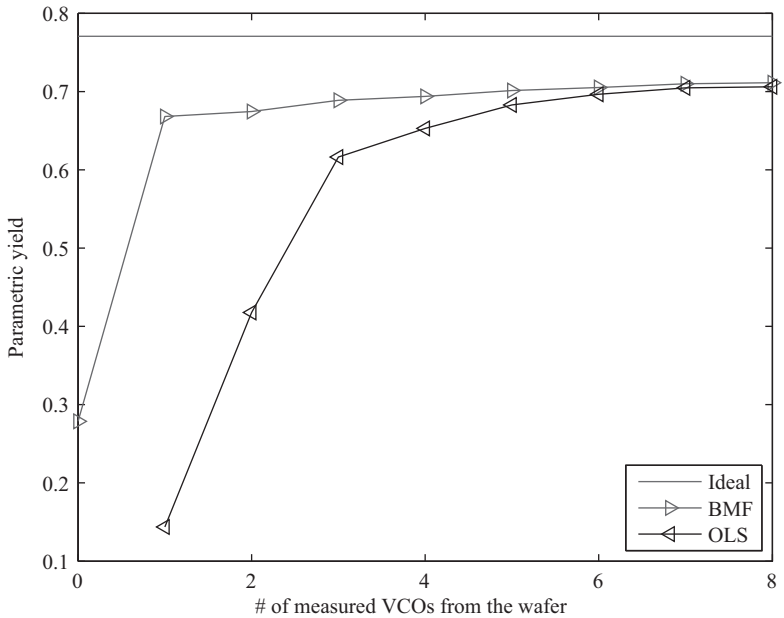


Figure 1.7 Post-self-healing parametric yield of the wafer is shown as a function of the number of measured VCOs from the wafer

Table 1.6 Measurement cost and parametric yield by self-healing

Self-healing method	# of measured VCOs	Parametric yield (%)
Ideal	61	77.05
OLS	4	65.30
BMF	1	66.80

- **OLS:** The traditional OLS method is applied to fit the indirect phase noise sensor based on a number of measured VCOs from the wafer. Next, the indirect sensor is applied to self-heal all the VCOs on the wafer.
- **BMF:** The indirect phase noise sensor learned by SR is considered as the early-stage model. Next, the proposed BMF algorithm is applied to calibrate the early-stage model and generate a late-stage model based on a few measured VCOs from the wafer. The late-stage model is then applied to self-heal all the VCOs on the wafer.

Figure 1.7 shows the parametric yield of the wafer achieved by three different self-healing methods given different number of measured VCOs from the wafer. Table 1.6 further summarizes the measurement cost for self-healing. Studying Figure 1.7 and

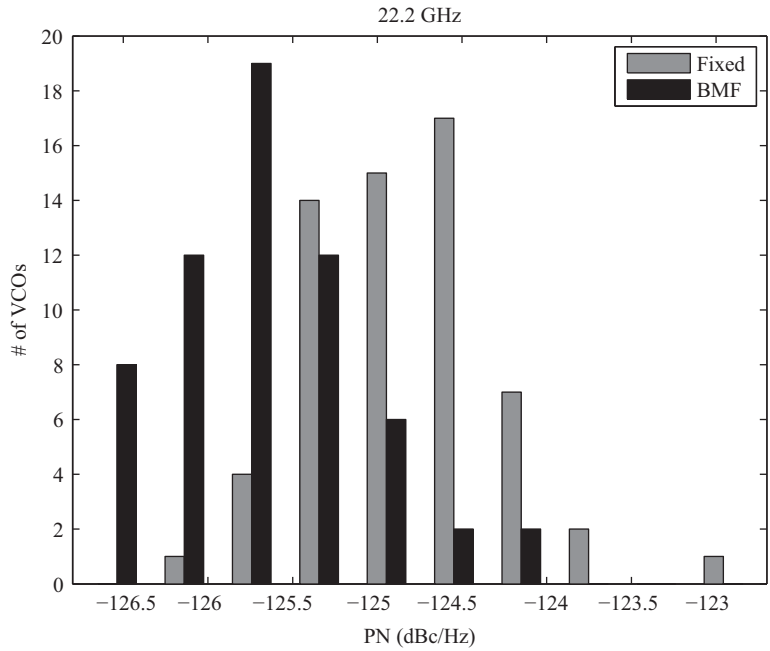


Figure 1.8 Histogram of the measured phase noise values from all the VCOs on the wafer at 22.2 GHz. Grey bars represent the results from Fixed where bias code is 4, and black bars represent the results from BMF where a single measured VCO is used from the wafer

Table 1.6 reveals several important observations. First, BMF requires substantially less number of measured VCOs to build the indirect phase noise sensor than the traditional OLS method. In this example, BMF needs to measure one VCO only, while OLS requires measuring four VCOs (4×) to achieve a similar yield.

Second, studying the BMF results in Figure 1.7, we notice that if no measurement data is collected from the wafer (i.e., the number of measured VCOs from the wafer is zero) and the self-healing is performed with the indirect sensor fitted from the early-stage data by SR, the post-self-healing parametric yield is only 27.87%. Once a single VCO is measured from the wafer, the indirect phase noise sensor is calibrated by BMF and the post-self-healing parametric yield is increased to 66.80%. It, in turn, demonstrates that the aforementioned model calibration is a critical step for yield enhancement.

Before ending this subsection, we compare the phase noise values from the proposed self-healing flow to those from the fixed bias voltage method (Fixed) to study why the proposed flow can achieve a much better parametric yield than Fixed. Figures 1.8–1.11 show the histograms of the measured phase noise values from all the VCOs at different frequencies. The grey bars show the results from Fixed where bias code is 4, and the black bars show the results from our described BMF technique

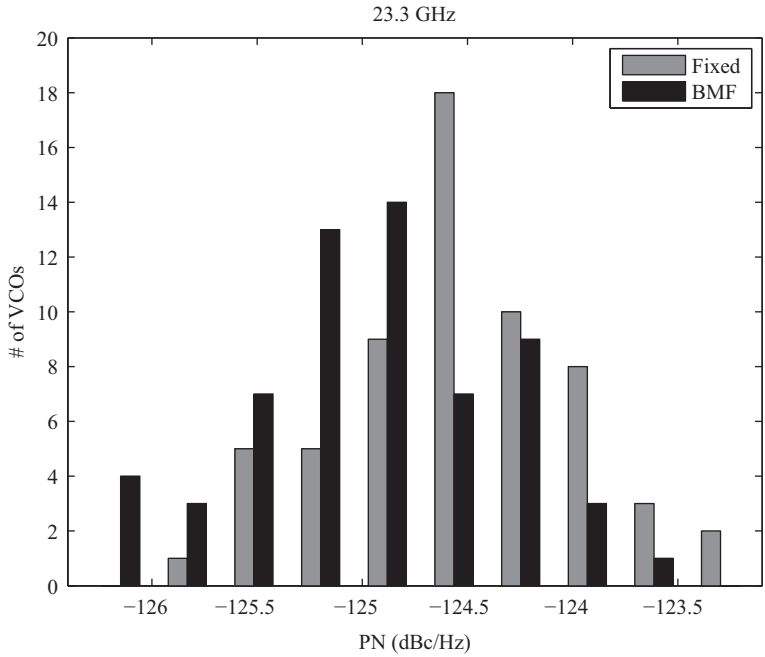


Figure 1.9 Histogram of the measured phase noise values from all the VCOs on the wafer at 23.3 GHz. Grey bars represent the results from Fixed where bias code is 4, and black bars represent the results from BMF where a single measured VCO is used from the wafer

with a single measured VCO from the wafer. From Figures 1.8–1.11, we have several observations. First, both BMF and Fixed get larger phase noise values at higher frequencies, which is consistent with our expectation. Hence, the phase noise specification at 26.2 GHz is the most difficult one to meet among all four phase noise specifications. Second, BMF technique can get much smaller phase noise values than Fixed at 26.2 GHz, which is the reason that BMF achieves a much better parametric yield than Fixed.

1.6.2 60 GHz LNA

The effects of process variations have become a challenging issue for RF LNA design and are even more significant at mm-wave frequencies. In particular, the gain, noise figure (NF) and matching of the LNA are susceptible to process variations. Figure 1.12 shows the schematic of a 60 GHz LNA designed in a 32 nm CMOS SOI process. The FETs, along with all the wire parasitics, were extracted from the layout to enable more accurate simulation. The circuit was simulated using high-frequency models

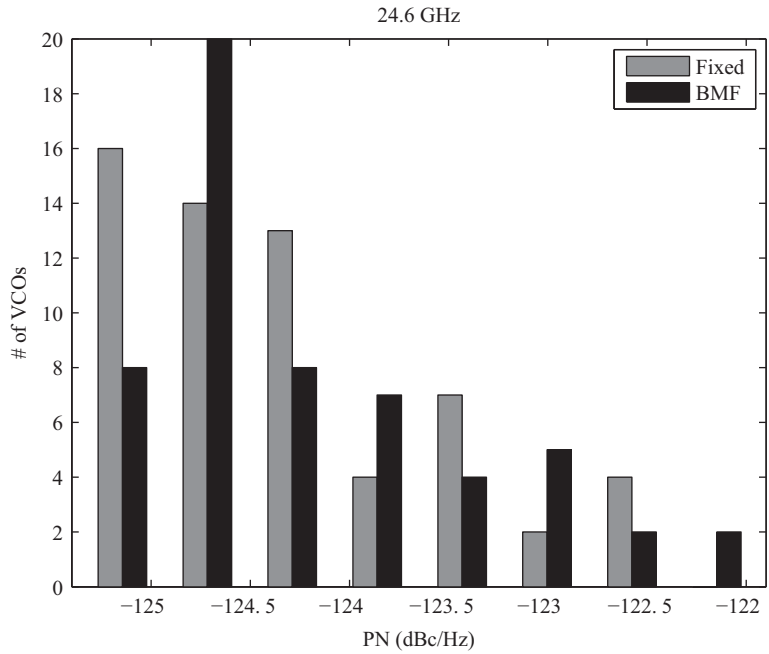


Figure 1.10 Histogram of the measured phase noise values from all the VCOs on the wafer at 24.6 GHz. Grey bars represent the results from Fixed where bias code is 4, and black bars represent the results from BMF where a single measured VCO is used from the wafer

for the transmission lines, capacitors, and resistors used in the design. Figure 1.13 shows NF sweeping vs. current biasing DAC code D_I at different temperatures. D_I directly controls I_S in Figure 1.12. Monte Carlo simulation results show that both gain (with mean value of 17.04 dB and standard deviation of 2.19 dB) and NF (with mean value of 5.15 dB and standard deviation of 0.49 dB) present large variability for this design. The variation of NF and gain of the LNA will significantly affect the performance of the whole receiver system. Therefore, it is essential to overcome the variations in the LNA to achieve low noise for the overall receiver. In this subsection, we assume that the gain of LNA will be measured by an on-chip peak detector, and we focus on NF self-healing only. More details about the LNA design can be found in Reference 20.

NF is generally difficult and expensive to measure directly on chip. Hence, here we apply the aforementioned indirect sensing technique, correlating NF with easy-to-measure PoMs. We collect a set of transistor-level Monte Carlo simulation data over the joint space of process, temperature T and bias current I_S. After the simulation data are collected, we apply the pre-silicon indirect sensor modeling procedure described

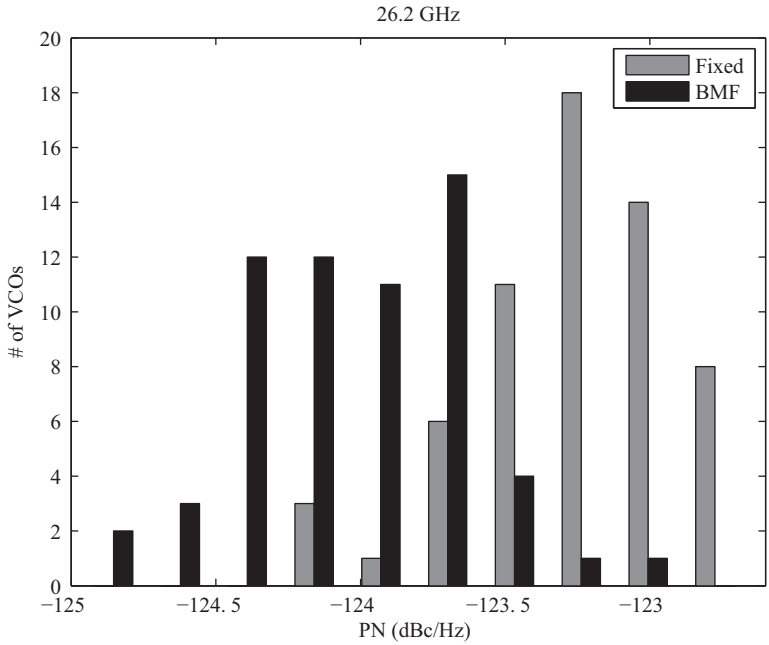


Figure 1.11 Histogram of the measured phase noise values from all the VCOs on the wafer at 26.2 GHz. Grey bars represent the results from Fixed where bias code is 4, and black bars represent the results from BMF where a single measured VCO is used from the wafer

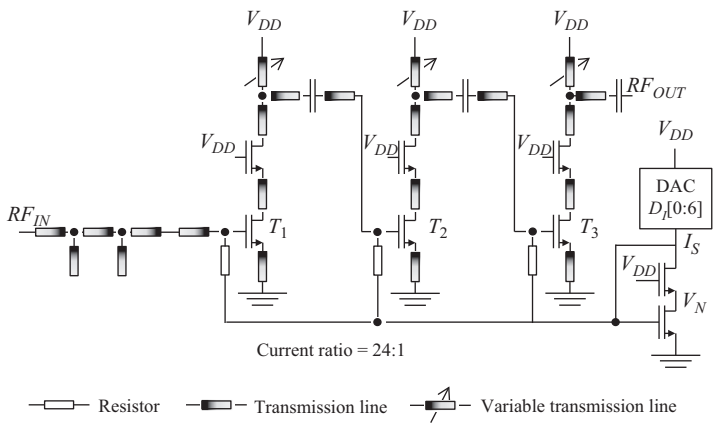


Figure 1.12 Simplified circuit schematic is shown for an LNA

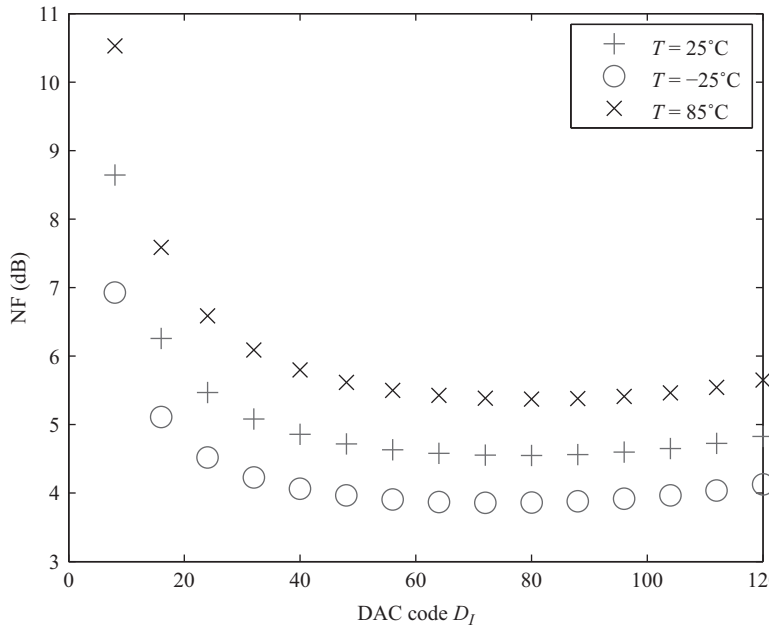


Figure 1.13 Simulated 60 GHz NF vs. current DAC code D_I at $T = 25°C$, $T = -25°C$, and $T = 85°C$

in Section 1.3. In this LNA example, there are multiple possible PoMs (e.g., peak detector output voltage, DC voltages, temperature) that can be correlated with NF. We need to mention here that the LNA chip in Figure 1.12 does not include a peak detector at 60 GHz. This is because the LNA will be integrated with a down-conversion mixer that will include a peak detector at the intermediate frequency which is easier to design. Here, we assume that the output signal amplitude is sensed by a "virtual" peak detector. In simulations, we record the 60 GHz output signal amplitude and use it as one of PoMs.

By applying SR algorithm, we are able to find the most important PoMs and select the important high-order terms associated with them. The final PoM set includes V_{PD} (the output voltage of the peak detector), V_N (the drain DC voltage in the biasing stage in Figure 1.12), D_I (the digital code controlling the bias current I_S in tunable current mirror), and T (temperature). Correlation coefficients between NF and V_{PD}, V_N, D_I, and T are -0.82, -0.81, -0.67, and 0.30, respectively. Here V_{PD} can be measured by a peak detector, V_N can be measured by an on-chip ADC, and T can be measured by an on-chip temperature sensor. D_I can be directly known from the digital code.

The indirect NF sensor model solved from SR is

$$\begin{aligned}\mathrm{NF}(V_{PD}, D_I, T, V_N) = {} & \alpha_0 + \alpha_1 \cdot V_{PD} + \alpha_2 \cdot D_I + \alpha_3 \cdot T + \alpha_4 \cdot V_{PD}^2 + \alpha_5 \cdot D_I^2 \\ & + \alpha_6 \cdot T^2 + \alpha_7 \cdot T^3 + \alpha_8 \cdot V_N^3 + \alpha_9 \cdot V_{PD} \cdot T + \alpha_{10} \cdot V_{PD}^3\end{aligned} \tag{1.40}$$

The mean error of this indirect sensor model is 0.170 dB. Using the indirect sensor model, we can predict NF according to on-chip sensor measurements. In this example, we apply the following self-healing flow:

1. Set bias current of all chips to minimum value.
2. V_{PD}, V_N, and T are measured using on-chip sensors.
3. Calculate predicted NF of each chip using the indirect sensor model and compare the result with the NF specification. If the NF meets the specification, the algorithm stops. Otherwise, the bias current is increased by a small value, after which steps 1 and 2 are repeated until the NF specification is met or the maximum DAC control word is reached.

The algorithm generally tries to find the minimum bias current that meets the NF specification.

Indirect sensor model error must be carefully considered during self-healing. Due to the NF prediction error, the estimated NF will be different from the actual NF. Therefore, to handle the uncertainty in NF prediction, a guard band is required. The guard band is the extra margin we leave for NF in self-healing, so that high yield can be achieved. The size of the guard band is determined by statistically modeling indirect sensor error. First, we collect the error data from indirect sensor model fitting. The error data are then fitted against a distribution by using kernel density estimation. The guard band can then be optimally determined once the error distribution is known. The calculated guard band is added to the predicted NFs for all chips to guarantee high yield.

To validate the self-healing algorithm with guard band at the simulation and design level, 40 chips are randomly generated from transistor-level simulations. After applying the self-healing procedure and adding guard band, the 40 chips achieve 100% yield with the NF specification of 5.5 dB. The average total LNA current for all chips is 14.7 mA. The histograms of NF and total current are shown in Figures 1.14 and 1.15. We also consider the fixed-biasing (Fixed) cases for comparison purpose. In the fixed-biasing cases, all the chips select the same tuning knob configuration. The power and yield of a set of fixed biasing cases are compared with the proposed self-healing method in Figure 1.16. The self-healing method is able to achieve 25% power reduction compared to the best fixed biasing case (with 19.4 mA total current), while not losing any yield. The key reason for the proposed self-healing method achieving better performance is that it adaptively selects an optimum bias current for each chip. For the chips with good NF, the algorithm will try to bias at low current so that power consumption is low. For the chips with bad NF, the algorithm tends to bias at a high current value so that the chip can meet the NF specification.

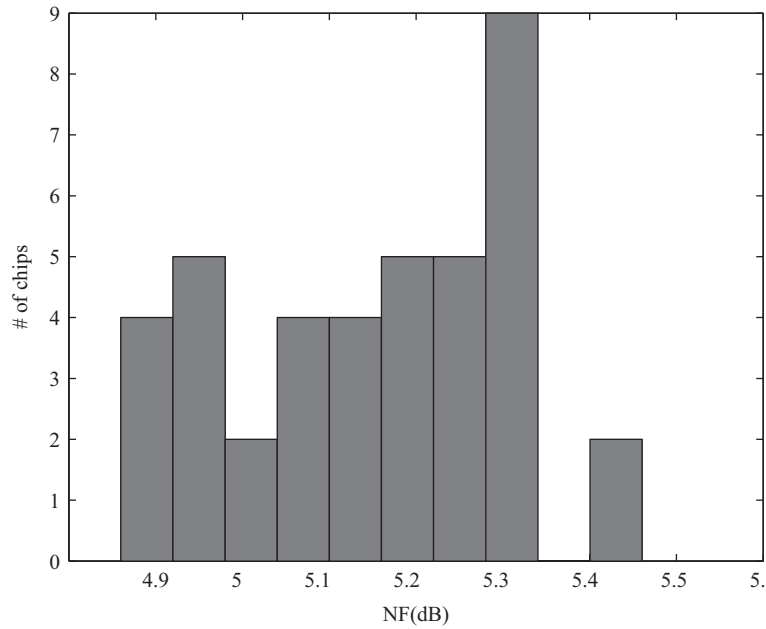


Figure 1.14 Histogram of NF after applying self-healing and adding guard band for 40 chips randomly generated from transistor-level simulations

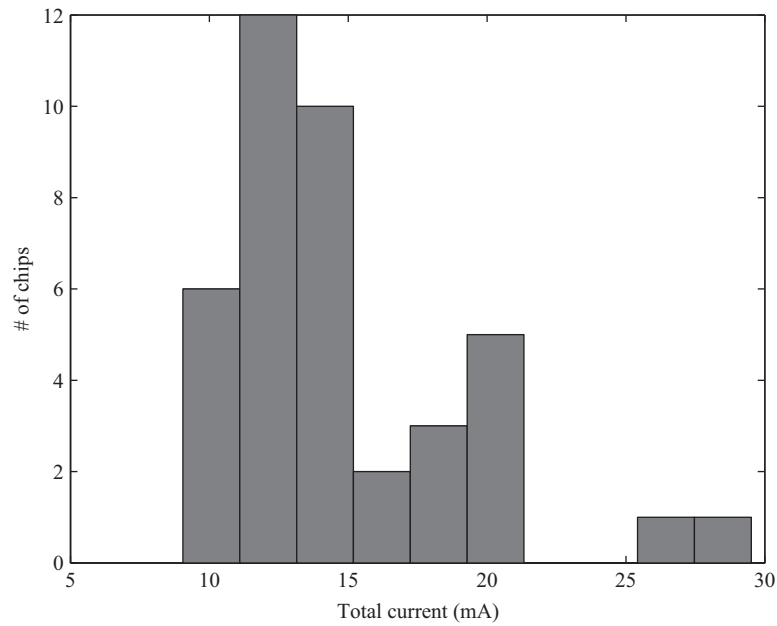


Figure 1.15 Histogram of total current after applying self-healing and adding guard band for 40 chips randomly generated from transistor-level simulations

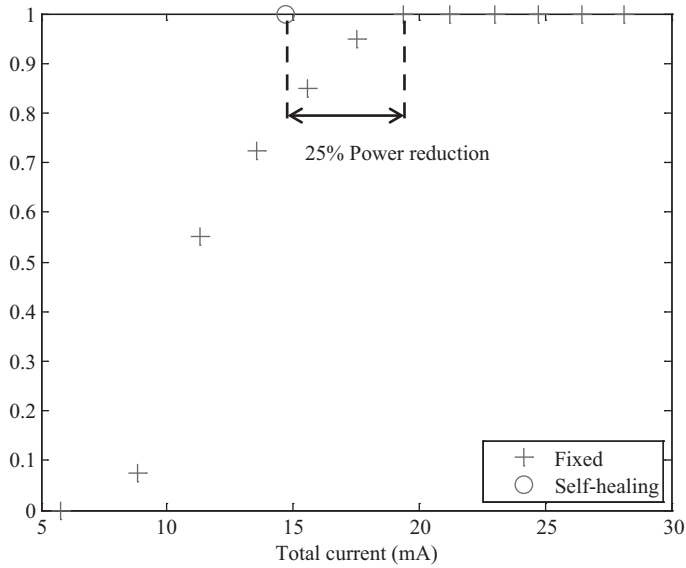


Figure 1.16 Comparison between self-healing and fixed

1.7 Conclusion

In this chapter, we describe a novel indirect performance sensing technique for on-chip self-healing of analog and RF circuits. In particular, a set of important basis functions are first identified by SR so that the overhead of on-chip self-healing can be minimized. Next, the indirect sensors are repeatedly calibrated by BMF to accommodate the process shift associated with manufacturing lines. The indirect sensors are eventually stored in an on-chip microcontroller to facilitate efficient on-chip self-healing. The proposed indirect performance sensing and on-chip self-healing methodology are validated by a 25 GHz differential Colpitts VCO and a 60 GHz LNA designed in a 32 nm CMOS SOI process. Our results show that the parametric yield and circuit performance are significantly improved after applying self-healing. In our future work, we will further extend the proposed indirect performance sensing and on-chip self-healing methodology to other large-scale circuits such as phase-locked loop.

References

[1] S. Nassif, "Modeling and analysis of manufacturing variations," in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 223–228, Sep. 2001.

[2] X. Li, J. Le, and L. Pileggi, *Statistical Performance Modeling and Optimization*. Delft, NL: Now Publishers, 2007.

[3] X. Li, P. Gopalakrishnan, Y. Xu, and L. Pileggi, "Robust analog/RF circuit design with projection-based performance modeling," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 26, no. 1, pp. 2–15, Jan. 2007.

[4] T. McConaghy, "High-dimensional statistical modeling and analysis of custom integrated circuits," in *Proc. IEEE Custom Integrated Circuits Conf.*, Sep. 2011.

[5] Semiconductor Industry Associate, *International Technology Roadmap for Semiconductors*, 2011.

[6] D. Han, S. Akbay, S. Bhattacharya, and A. Chatterjee, "On-chip self-calibration of RF circuits using specification-driven built-in self test (S-BIST)," in *Proc. IEEE Int. On-Line Testing Symp.*, pp. 106–111, July 2005.

[7] H. Stratigopoulos, S. Mir, E. Acar, and S. Ozev, "Defect filter for alternate RF test," in *Proc. IEEE European Test Symp.*, pp. 101–106, May 2009.

[8] E. Acar and S. Ozev, "Low-cost characterization and calibration of RF integrated circuits through *I–Q* data analysis," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 28, no. 7, pp. 993–1005, July 2009.

[9] D. Han, B. Kim, and A. Chatterjee, "DSP-driven self-tuning of RF circuits for process-induced performance variability," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 2, pp. 305–314, Feb. 2010.

[10] K. Jayaraman, Q. Khan, B. Chi, W. Beattie, Z. Wang, and P. Chiang, "A self-healing 2.4 GHz LNA with on-chip S11/S21 measurement/calibration for in-situ PVT compensation," in *Proc. IEEE Radio Frequency Integrated Circuits Symp.*, pp. 311–314, May 2010.

[11] N. Kupp, H. Huang, P. Drineas, and Y. Makris, "Post-production performance calibration in analog/RF devices," in *Proc. IEEE Int. Test Conf.*, Nov. 2010.

[12] V. Nataranjan, S. Sen, A. Banerjee, and A. Chatterjee, "Analog signature-driven postmanufacture multidimensional tuning of RF systems," *IEEE Des. Test Comput.*, vol. 27, no. 6, pp. 6–17, Dec. 2010.

[13] D. Howard, P. Saha, S. Shankar, R. Diestelhorst, T. England, and J. Cressler, "A UWB SiGe LNA for multi-band applications with self-healing based on DC extraction of device characteristics," in *Proc. IEEE Bipolar/BiCMOS Circuits and Technology Meeting*, pp. 111–114, Oct. 2011.

[14] A. Goyal, M. Swaminathan, A. Chatterjee, D. Howard, and J. Cressler, "A new self-healing methodology for RF amplifier circuits based on oscillation principles," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 10, pp. 1835–1848, Oct. 2012.

[15] C. Chien, A. Tang, F. Hsiao, and M. Chang, "Dual-control self-healing architecture for high performance radio SoC's," *IEEE Des. Test Comput.*, vol. 29, no. 6, pp. 40–51, May 2013.

[16] B. Sadhu, M. Ferriss, A. Natarajan, *et al.*, "A linearized low-phase-noise VCO-based 25-GHz PLL with autonomic biasing," *IEEE J. Solid-State Circuits*, vol. 48, no. 5, pp. 1138–1150, May 2013.

[17] M. Sanduleanu, A. Valdes-Garcia, Y. Liu, *et al.*, "A 60 GHz, linear, direct down-conversion mixer with mm-wave tunability in 32 nm CMOS SOI," in *Proc. IEEE Custom Integrated Circuits Conf.*, Sep. 2013.

[18] S. Sun, F. Wang, S. Yaldiz, *et al.*, "Indirect performance sensing for on-chip self-healing of analog and RF circuits," *IEEE Trans. Circuits Syst. I*, vol. 61, no. 8, pp. 2243–2252, Aug. 2014.

[19] D. Howard, P. Saha, S. Shankar, *et al.*, "A SiGe 8–18-GHz receiver with built-in-testing capability for self-healing applications," *IEEE Trans. Microw. Theory Tech.*, vol. 62, no. 10, pp. 2370–2380, Oct. 2014.

[20] J. Plouchart, B. Parker, B. Sadhu, *et al.*, "Adaptive circuit design methodology and test applied to millimeter-wave circuits," *IEEE Des. Test*, vol. 31, no. 6, pp. 8–18, Dec. 2014.

[21] P. Variyam, S. Cherubal, and A. Chatterjee, "Prediction of analog performance parameters using fast transient testing," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 3, pp. 349–361, Aug. 2002.

[22] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Stat. Soc. Ser. B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[23] X. Li, "Finding deterministic solution from underdetermined equation: large-scale performance modeling of analog/RF circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 29, no. 11, pp. 1661–1668, Nov. 2010.

[24] C. Bishop, *Pattern Recognition and Machine Learning*. New York, US: Springer, 2006.

[25] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2009.

[26] E. Candes, "Compressive sampling," *Int. Congress of Mathematicians*, 2006.

[27] D. Donoho, "Compressed sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[28] J. Tropp and S. Wright, "Computational methods for sparse solution of linear inverse problems," *Proc. IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[29] D. Donoho and J. Tanner, "Precise undersampling theorems," *Proc. IEEE*, vol. 98, no. 6, pp. 913–924, 2010.

[30] G. Golub and C. Loan, *Matrix Computations*. Baltimore, USA: Johns Hopkins University Press, 1996.

[31] F. Wang, W. Zhang, S. Sun, X. Li, and C. Gu, "Bayesian model fusion: large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data," in *Proc. Des. Automat. Conf.*, 2013.

[32] J. Plouchart, M. Ferriss, A. Natarajan, *et al.*, "A 23.5 GHz PLL with an adaptively biased VCO in 32 nm SOI-CMOS," *IEEE Trans. Circuits Syst. I*, vol. 60, no. 8, pp. 2009–2017, Aug. 2013.

[33] D. Leeson, "A simple model of feedback oscillator noise spectrum," *Proc. IEEE*, vol. 54, no. 2, pp. 329–330, Feb. 1966.

Chapter 2

On-chip gate delay variability measurement in scaled technology node

Bishnu Prasad Das[1], *Bharadwaj Amrutur*[2] *and Hidetoshi Onodera*[3]

The previous chapter focused on the process variations issue in the analog circuits. This chapter focuses on the issue of process variations in the digital integrated circuits. The chapter first presents detailed discussions of the sources of variability and types of variability. Then the discussions are focused on the impact of process variations on the propagation delay of the digital integrated circuits. Test chip results have been discussed thoroughly as an experimental evaluation.

2.1 Introduction

With the advancement of nanoscale fabrication process, it is possible to fabricate transistors with its length in sub-10 nm in silicon which enables packing of billions of transistor in a single die. However, it is extremely difficult to fabricate these tiny transistors with well-defined characteristics and this leads to variation in circuit performance parameters such as delay and power [1]. The variations can be due to process and environment. The process variation is due to the fluctuation of attributes of the transistor such as length, oxide thickness and doping density during the fabrication of the chip. It is typically classified as within-die, die-to-die, wafer-to-wafer and lot-to-lot. In the past, the die-to-die variations were significant. However, in the scaled technology nodes, the contribution of variation due to within-die is becoming significant. The major sources of within-die variation are line-edge-roughness (LER), oxide-thickness variation (OTV) and random dopant fluctuation (RDF). The noticeable increase in within-die variation has brought to light the need to design on-chip circuit structures to measure these variations such that it can be incorporated into the transistor models.

[1] Indian Institute of Technology, Roorkee, India
[2] Indian Institute of Science, Bangalore, India
[3] Kyoto University, Kyoto, Japan

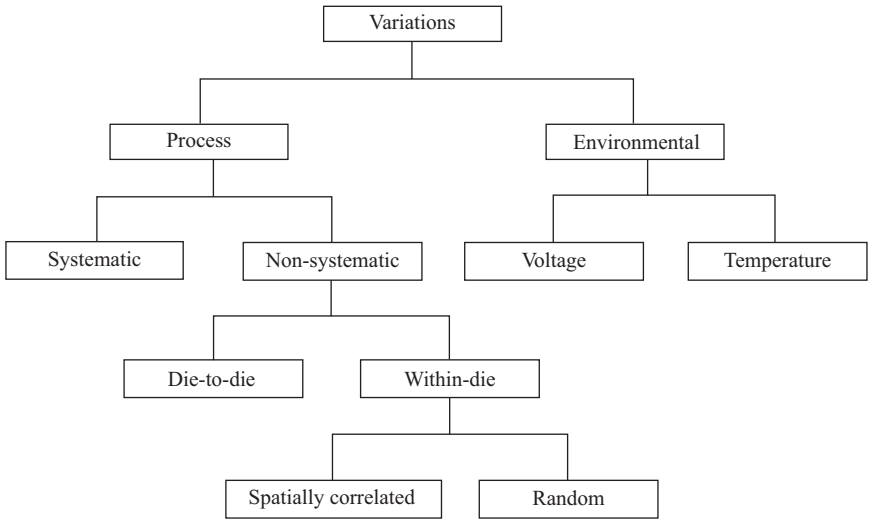


Figure 2.1 Taxonomy of variation

Process variation measurement is an essential component of integrated circuit design flow during the early stage of process development. On-chip test structures enable accurate transistor-level model creation with the measured silicon data. Exploiting silicon data while creating Simulation Program with Integrated Circuit Emphasis (SPICE) model reduces pessimism in the design in the later stage of product development. In this chapter, some of the variability measurement techniques are described.

2.2 Classification of variability

Figure 2.1 shows the types of variation in Very Large Scale Integration (VLSI) circuits [2]. The variability can mainly be categorized as

1. **Process variation**
   The main sources of random process variation are RDF, LER and OTV. This leads to variation in threshold voltage and hence in drive current. Hence, the performance parameters such as delay and power will deviate from their desired nominal values.
2. **Environmental variation**
   The environmental variations include supply voltage and temperature variations. The source of supply variation is due to di/dt noise and voltage drop in the power grid. The variation in switching activity due to non-uniform distribution of transistors across the chip leads to temperature hot spots inside the chip.

The process variation can further be classified as systematic and non-systematic based on ease of predictability.

1. **Systematic variation**
   The physical parameter variations which is well understood and can be predicted *a priori* by analyzing the layout of the design. Examples are variations due to optical proximity, Chemical Mechanical Polishing (CMP), metal fill, etc. These variations can be modeled before manufacturing by detailed analysis of the layout.
2. **Non-systematic variation**
   These process variations cannot be predicted by just analyzing the design or layout. These arise from the causes that are independent of design implementation. However, the statistical characteristics of these process parameters can be determined and used during the design time. The examples are variations due to RDF, LER and OTV.

It is common practice in design flow to model systematic and non-systematic variations statistically. In the advanced stage of the design process, after gaining more information, the systematic components of variation are modeled deterministically.

Depending on the spatial scales of the variation, the non-systematic variation has following two categories:

1. **Die-to-die or inter-die or global variation**
   Die-to-die (or inter-die or global) variations affect all transistors within a die in the same way and have identical effect. For example, gate length variation of all the devices on the same chip will be larger or smaller less than the nominal value. The die-to-die variations result in shift in the process from lot to lot, wafer to wafer, reticle to reticle on each gate in the chip.
2. **Within-die or intra-die or local variation**
   Within-die (or intra-die or local) variations affect each device within the same die differently. For example, some devices on a die have smaller gate length whereas other devices on the same die have a larger gate length.

The within-die variations can be classified as spatially correlated and random variations.

1. **Spatially correlated variations**
   Within-die variations which exhibit similar characteristic for devices in small neighborhood in the die than those placed far apart are called spatially correlated variations.
2. **Random or independent variations**
   Within-die variations which is statistically independent from other device variations are called random or independent variation. For example, RDF and LER are two major random variations. With the continuous process scaling, the contributions of random component of variation are increasing.

If X is the delay of a logic gate, then

$$X = X_0 + \Delta X_g + \Delta X_l^s + \Delta X_l^r \tag{2.1}$$

where X_0 is the nominal delay of the gate, ΔX_g is the global delay variation, ΔX_l^s is the spatially correlated local delay variation and ΔX_l^r is the random local delay variation [3]. The two components ΔX_g and ΔX_l^s are shared by many gates within a small region with spatial correlation and hence standard ring oscillator (RO)-based measurement techniques suffice. The final component ΔX_l^r is unique to each gate and hence needs to be measured at the level of a single gate.

2.3 Sources of variability

2.3.1 Random dopant fluctuations

In the past, the transistor's threshold voltage variation due to the RDF was one of the serious challenges for analog designers. However, in the scaled technology nodes, it also becomes a headache for digital designers and serves as a performance and yield limiter in digital designs. As channel length decreases in each technology generation, the number of dopant atoms reduces in the channel depletion region [4]. In technology having channel length above 1 μm, a few thousands of dopant atoms are available in the channel area leading to negligible deviation in dopant atoms. However, in scaled technology nodes, the number of dopant atoms is around 10–100 atoms, leading to significant fluctuations in the dopant atoms in the channel area.

The analytical expression of dependence of σV_{th} on RDF for planar devices was established in References 5, 6 and stated in (2.2). It shows that σV_{th} has linear relationship with the oxide thickness T_{ox}, an inverse square-root relationship with the product of effective length and width (L_{eff} and W_{eff}) and a fourth-root relationship with N_{tot} (i.e., the total doping density).

$$\sigma V_{th} = \frac{\sqrt[4]{4q^3\varepsilon_{si}\phi_B}}{2} \cdot \frac{T_{ox}}{\varepsilon_{ox}} \cdot \frac{\sqrt[4]{N_{tot}}}{\sqrt{L_{eff}W_{eff}}} \tag{2.2}$$

According to famous Pelgram's model [7], σV_{th} increases as the channel length decreases in the scaled technology nodes.

$$\sigma V_{th} = \frac{A_{vth}}{\sqrt{WL}} \tag{2.3}$$

where A_{vth} is the slope of the plot of σV_{th} versus $1/\sqrt{WL}$.

2.3.2 Line edge roughness

LER is typically caused by the random fluctuation of line edge of the transistor from its desired value. The major source of LER variation include photo-resist LER due

to type and thickness of resist and substrate reflectivity, image contrast and gate poly etching conditions such as size of poly grain and its doping [8].

The measured LER variation from various lithography processes shows that it does not scale with linewidth or the critical dimension scaling as reported by International Technology Roadmap for Semiconductors (ITRS) road map. Its value stays typically around 5 nm as mentioned in Reference 9. Authors in Reference 9 explored the combined impact of LER and random dopants on current fluctuations and demonstrated that the both the sources of variations are statistically independent for channel length of the transistor greater than 30 nm. So the effective standard deviation σ_t is given as follows:

$$\sigma_t = \sqrt{\sigma_{\mathrm{RDF}}^2 + \sigma_{\mathrm{LER}}^2} \tag{2.4}$$

where σ_{RDF} and σ_{LER} are the standard deviation of drain current due to the RDF and LER, respectively. The LER-induced channel fluctuation depends strongly on the channel length and would be the dominant source of variations compared to random fluctuation due to short channel effects.

In order to increase the device performance, the minimum channel length is chosen in the region of steep Vth roll-off in the graph of length versus threshold voltage. Due to this choice, the channel length variation results in severe variation in threshold voltage. Hence, it is wise to increase the channel length beyond the Vth roll-off point to reduce the threshold voltage variation where the performance is not critical.

2.3.3 Oxide thickness variation

In the advanced scaled technology node, the thickness of the oxide layer scales down to a few layers of silicon atoms which leads to significant increase in interface roughness. This effect would produce significant amount of OTV within the gate portion of identically drawn transistors. The OTV due to interface roughness would lead to variation of mobility, gate tunneling current and eventually the threshold voltage of each transistor in the chip.

The simulation results from Reference 10 show that threshold voltage variation due to OTV and RDF is statistically independent. The total standard deviation is given by

$$\sigma V_{\mathrm{th}}^{\mathrm{t}} = \sqrt{\left(\sigma V_{\mathrm{th}}^{\mathrm{RDF}}\right)^2 + \left(\sigma V_{\mathrm{th}}^{\mathrm{OTV}}\right)^2} \tag{2.5}$$

where $\sigma V_{\mathrm{th}}^{\mathrm{RDF}}$ and $\sigma V_{\mathrm{th}}^{\mathrm{OTV}}$ are the standard deviation of threshold voltage due to the RDF and OTV, respectively.

2.4 Related work on variability measurement

2.4.1 Gate delay variability

Several works [11–15] have been reported to measure the random local variability of static or DC parameters such as threshold voltage, oxide thickness, etc. All-digital

process monitor has been proposed by Klass *et al.* [11] which converts process variability into digital code. A technique to measure local threshold voltage variation is reported by Rao *et al.* [12]. A sense amplifier-based test structure [13] has been used to measure threshold voltage mismatch between two closely spaced transistors. A test structure to monitor negative-bias-temperature-instability (NBTI) and oxide degradation is proposed by Karl *et al.* [14]. The direct measurement of dynamic parameter such as delay provides enough information which closely match real-time designs compared to static or DC parametric measurements. Hence, we discussed circuit techniques to measure delay of individual gate to study the impact random variation on delay.

ROs are typically employed to measure the impact of delay variation caused by process variations. These test structures can be easily implemented on-chip and they have strong sensitivity to process parameter fluctuations [16–21]. The local random component of delay is averaged out in a RO with large number of stages. So, it is hard to obtain random gate delay variation using a simple RO. The averaging of random variation can be minimized by implementing ROs with 3–5 stages; however, the frequency of oscillation will be extremely high which will lead to difficulty in realization. Onodera [19] showed the variability in three technology nodes and explained the importance of within-die variability using on-chip transistor-level and RO-based measurements.

Delay locked loop (DLL)-based test structure [22] has been reported to measure the delay of a gate. However, requirement of phase detector and charge pump in DLL test structure increases area overhead which is not acceptable especially for large-scale implementation which is required for variability characterization.

In picosecond imaging circuit analysis (PICA) [23, 24] method, the delay of individual gate can be measured by counting the number of intra-red (IR) photons from the backside of a thinned package chip. However, the requirement of IR source in measurement process makes it an expensive approach.

The random sampling technique in Reference 25 has been reported to measure the delay of standard cells. In this approach, a periodic pulse signal is applied to the input of the gate under test (GUT) and then the sampling of input and output waveform is done randomly. The gate delay is estimated using the joint signal probability of the samples. The rise and fall delays of the individual gate can be measured at different input slew and output load conditions using this approach. However, the random local variation [26] affects the two samplers differently leading to inaccurate delay measurement. The variability of samplers can be reduced by using wider transistor size which would increase the area of the test structure, as large-scale implementation is required for random local variability characterization. Another demerit of the approach is that wider transistor used in sampler will increase the output loading of the GUT. Hence, it would be hard to measure the delay under low-load condition.

The ROs with the modified cell as shown in Figure 2.2 is proposed in Reference 27 to measure the incremental delay difference between the two propagation delays of the same cell for two different control conditions. The major disadvantage of this technique is that the modified cell is not used in real designs, as it is not available

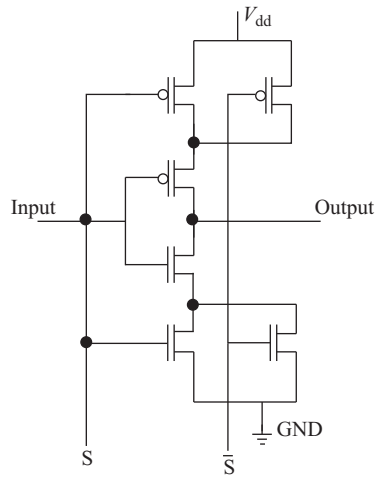


Figure 2.2 Modified inverter schematic to measure within-die variation [28]. © 2009 IEEE

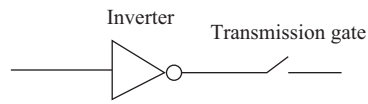


Figure 2.3 Modified inverter schematic to measure within-die variation in Reference 28. © 2009 IEEE

in standard cell library. Additionally, the modified cell cannot measure systematic spatially co-related process variation.

The cell-to-cell delay mismatch due to process variations is proposed in Reference 29. The type of cell used in this method is shown in Figure 2.3. The importance of symmetry in the RO-based method is discussed in Reference 29. However, as the multiplexer is used outside the RO, the layout of the test structure is not regular.

2.4.2 Rise and fall gate delay variability

The on-chip measurement circuits are broadly categorized into two types such as digital and analog. The digital techniques are preferred over analog techniques due to easy on-chip implementation and less calibration requirements. RO-based measurement [28, 30] circuits are most popular digital techniques for on-chip variability and/or reliability measurement. Some of the analog on-chip measurement circuits are random sampling [25] and on-chip sampling oscilloscope [31]. In Reference 25, the random sampling is applied to the input and output of a periodic waveform which is supplied to the input of the GUT. The rise/fall gate delay is estimated from the joint probability distribution of the samples. In Reference 31, the rise/fall gate delay is measured using an on-chip sampling oscilloscope which consists of sampling timing

generator, reference voltage generator and sampling head. However, these techniques need analog circuits such as reference voltage generator, sampling head [31] and sample-hold circuits [25] which requires a lot of calibration and post-processing.

2.5 Gate delay measurement using reconfigurable ring oscillator

2.5.1 Gate delay measurement cell

The basic concept of gate delay measurement technique is illustrated in Figure 2.4 which has two paths from input to output. Conceptually, the delay of the GUT is determined by taking the difference of delay between the two paths. This measurement is quite similar to sensitivity analysis in SPICE. The schematic of the gate delay measurement cell (GDMC) is shown in Figure 2.5 [32, 33] and consists of the GUT, two identical multiplexers and four inverters. The delay of the GUT I_1 is measured using this technique. The other inverters I_2, I_3, I_4 and I_5 are basically utilized for buffering and load balancing. Two different configuration settings of multiplexer Mux_1 enable the calculation of gate delay by taking the difference of two period measurements of the RO. Finally, the calculated delay will be the sum of the delay of GUT and the difference in path delays between the two inputs and output of the multiplexer Mux_1 (i.e., between input A to output Y and input B to output Y) due to

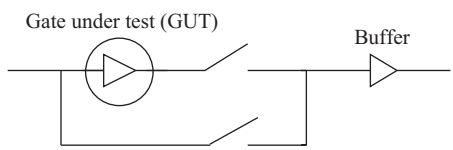


Figure 2.4 Concept of gate delay measurement [28]. © 2009 IEEE

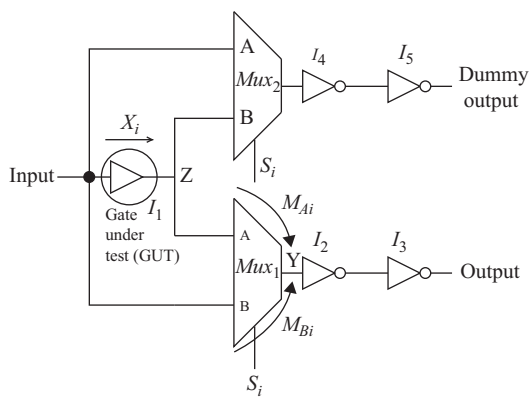


Figure 2.5 Schematic of a GDMC [28]. © 2009 IEEE

unequal slew input to two inputs of Mux_1 for the two different multiplexer settings. The accuracy of gate's delay measurement is limited by the mismatch in multiplexer Mux_1. There are two reasons for mismatch in multiplexer Mux_1: (1) random local mismatch in the multiplexer transistors and (2) input slew variations at the inputs of the multiplexer. The proper sizing of the multiplexers and buffering can reduce both these mismatches.

In each GDMC, the two inputs A and B of the multiplexer Mux_1 are connected to the output of GUT I_1 and the input of the GUT I_1, respectively. The output Y of the multiplexer Mux_1 is connected to buffers which is implemented using two inverters I_2 and I_3. It helps in reducing the input slew variation of the signal occurred due to configuration change in the next stage. Multiplexer Mux_2 and inverters I_4, I_5 are dummy structures used to provide equal loading to inverter I_3 of the previous stage irrespective of status of the select input of the multiplexers. If $S_i = 0$, the inverter I_3 of the previous stage is driven by the combined load at pin "A" of Mux_2 and GUT I_1. If $S_i = 1$, the inverter I_3 of the previous stage is driven by the combined load at pin "B" of Mux_1 and GUT I_1. Hence, the delay of the inverter I_3 of the previous stage does not depend upon states of S_i.

Symmetric multiplexer of large size which has balanced delays between input A to output Y and input B to output Y are used to reduce any systematic mismatches. The sizes of multiplexer Mux_1, inverters I_2, I_3 are same as of multiplexer Mux_2, inverters I_4, I_5, respectively. The multiplexers Mux_1 and Mux_2 are matched in the layout. The size of the inverters I_3 and I_5 is same as the output stage of the GUT I_1. The above structure allows symmetry and load matching which helps in finding the individual gate delay.

2.5.2 Reconfigurable ring oscillator structure

The RO consisting of the GDMC is shown in Figure 2.6. As the number of stages of the RO can be configured externally, hence it is known as reconfigurable ring oscillator (RRO). The basic idea of gate delay measurement using RRO is applicable for the measurement of both a non-inverting as well as an inverting logic gate. As odd number of inverting stages are essential for the oscillation of RRO, there is an additional requirement for selecting the configuration of multiplexer such that odd number of inverting stages will be in the path. By properly selecting the configuration of the multiplexer, it is possible to calculate the delay of any logic gate and is described in detail in the next sub-section.

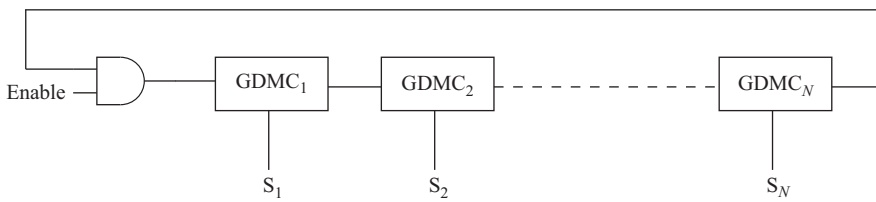


Figure 2.6 RO based on GDMC [28]. © 2009 IEEE

For clarity of explanation, a two-stage and a five-stage RROs are leveraged to explain the non-inverting and inverting gate delay measurements, respectively. However, RRO-based gate delay measurement can easily be extended for any number of stages of RRO.

2.5.2.1 Non-inverting average gate delay measurement

A two-stage RRO is used as shown in Figure 2.7 to explain the non-inverting gate delay measurement. Table 2.1 shows the notations and definition of variables needed for gate delay measurement. All the delays in Table 2.1 are the average of rise and fall delays. As the inverters I_4 and I_5 are not in the path of the RO, they do not contribute to the frequency of the RRO.

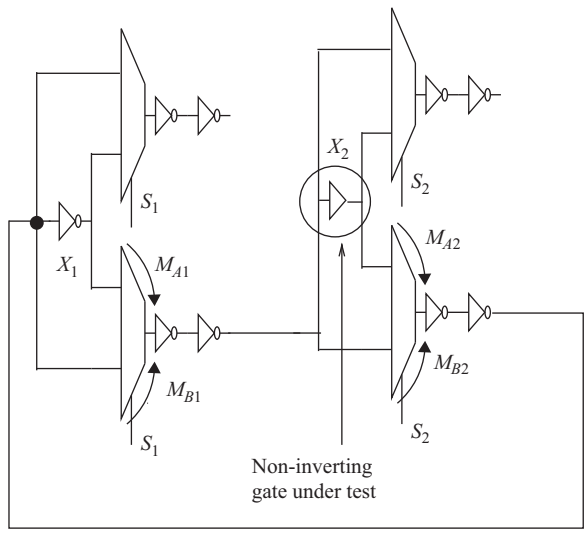


Figure 2.7 Non-inverting gate delay measurement showing buffer under test [28]. © 2009 IEEE

Table 2.1 Notations and definitions of variables for gate delay measurement

Notations	Definitions of variables
X_i	Average of rise and fall delays of the ith gate I_1 under test
M_{Ai}	Delay between inputs A to output Y of the multiplexer Mux_1
M_{Bi}	Delay between input B to output Y of the multiplexer Mux_1
K	Delay of the rest of the elements of the RO like the buffers consisting of inverters I_2, I_3 of all the cells, the wires, etc.
S	Status or configuration vector of the RRO

The technique of delay measurement of non-inverting gate is as follows. Let us consider a non-inverting gate (e.g., buffer) is placed in second stage instead of GUT I_1 (refer Figure 2.5). The minimum number of stages of GDMC needed for non-inverting gate delay measurement is 2. Let S be the status vector consisting of 2 bits. If $S_i = 0$, then input A to output Y of the multiplexer Mux_1 of the ith stage GDMC is selected. If $S_i = 1$, then input B to output Y of the multiplexer Mux_1 of the ith stage GDMC is selected. T_1 and T_2 are the period of RO for two status vectors $S = 00, 01$, respectively. Since each GDMC switches twice during a complete cycle, twice the sum of all the average cell delays equates to a period of the RO signal.

When select status vector $S = 00$, then the input A to output Y of stages 1 and 2 are connected.

$$2 * \left(\sum_{i=1}^{2} (X_i + M_{Ai}) + K \right) = T_1 \tag{2.6}$$

When select status vector $S = 01$, then the input A to output Y of stage 1 is connected and the input B to output Y of stage 2 is connected.

$$2 * ((X_1 + M_{A1}) + M_{B2} + K) = T_2 \tag{2.7}$$

Taking the difference of (2.6) and (2.7), we get

$$X_2 + (M_{A2} - M_{B2}) = (T_1 - T_2)/2 \tag{2.8}$$

Period measurements of the RO $T_1 - T_2$, with two different control word settings lead to (2.8) for the delay of buffer I_1 in the second stage along with the residual error term, which is the difference in delay from input A to output Y and from input B to output Y of the multiplexer Mux_1.

2.5.2.2 Inverting average gate delay measurement

Five-stage RRO is used as shown in Figure 2.8 to explain the delay measurement circuit for an inverting gate (e.g., inverter). The status vector S, X_i, M_{Ai} and M_{Bi} and K are defined in Table 2.1. The minimum number of stages of GDMC needed for inverting gate delay measurement is 5. Hence, the status vector is 5 bits. T_3, T_4, T_5 and T_6 are the period of RRO for four status vectors $S = 00000, 00011, 00110, 00101$, respectively.

When select status vector $S = 00000$, then the input A to output Y of stages 1, 2, 3, 4 and 5 are connected. Then,

$$2 * \left(\sum_{i=1}^{5} (X_i + M_{Ai}) + K \right) = T_3 \tag{2.9}$$

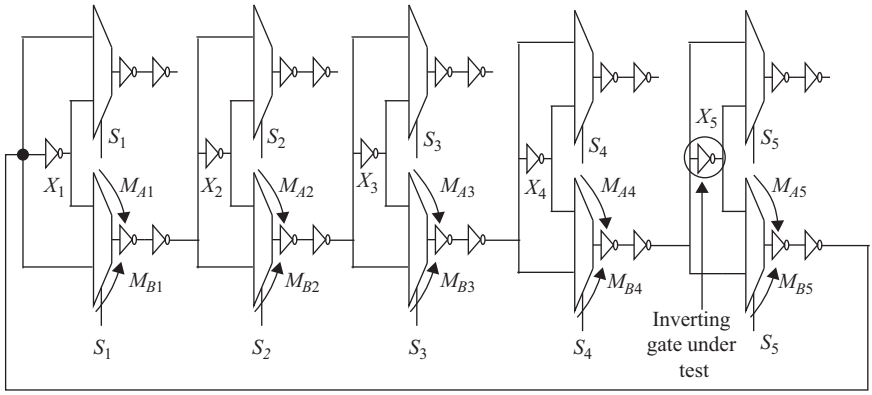


Figure 2.8 Inverting gate delay measurement showing inverter under test [28]. © 2009 IEEE

When select status vector $S = 00011$, then the input A to output Y of stages 1, 2 and 3 are connected and the input B to output Y of stages 4 and 5 are connected.

$$2 * \left(\sum_{i=1}^{3} (X_i + M_{Ai}) + \sum_{j=4}^{5} M_{Bj} + K \right) = T_4 \tag{2.10}$$

Taking the difference between (2.9) and (2.10), we get

$$X_4 + M_{A4} + X_5 + M_{A5} - M_{B4} - M_{B5} = (T_3 - T_4)/2 \tag{2.11}$$

When select status vector $S = 00110$, then the input A to output Y of stages 1, 2 and 5 are connected and the input B to output Y of stages 3 and 4 are connected.

$$2 * \left(\sum_{i=1,2,5} (X_i + M_{Ai}) + \sum_{j=3}^{4} M_{Bj} + K \right) = T_5 \tag{2.12}$$

When select status vector $S = 00101$, then the input A to output Y of stages 1, 2 and 4 are connected and the input B to output Y of stages 3 and 5 are connected.

$$2 * \left(\sum_{i=1,2,4} (X_i + M_{Ai}) + \sum_{j=3,5} M_{Bj} + K \right) = T_6 \tag{2.13}$$

Taking the difference between (2.12) and (2.13), we get

$$-X_4 - M_{A4} + X_5 + M_{A5} + M_{B4} - M_{B5} = (T_5 - T_6)/2 \tag{2.14}$$

Adding (2.11) and (2.14), we get

$$X_5 + (M_{A5} - M_{B5}) = (T_3 - T_4 + T_5 - T_6)/4 \tag{2.15}$$

To find the delay of inverter I_1 in the fifth stage along with the residual error term requires four period measurements of RRO (i.e., T_3–T_6), using the four different control words. The residual term is due to the difference of delay between input A to output Y and input B to output Y delays of the multiplexer Mux_1 of fifth stage. The two components of residual error are (i) unequal slew at the two inputs A and B of the multiplexer Mux_1 and (ii) local delay variation between input A to output Y and input B to output Y of multiplexer Mux_1. The unequal slew rate at the two inputs of Mux_1 for the two different multiplexer settings leads to a residual delay error between the inputs A, B and multiplexer output Y. We estimated this error by performing SPICE simulation of parasitic extracted layout in the absence of process variations and is found to be 1.2 ps. As the approach requires the computation of difference of linear delay equations, the residual delay error term for the multiplexer delay mismatch cancels out the global and spatially correlated variation component leaving behind only the random local variation term. Note that X_5 includes sum of nominal gate delay, global and all the local variation terms (refer to (2.1)).

Measuring the delay of two inverters require four different settings of the control words. Since this technique of measurement involves computation of difference of linear equations, the impact of systematic error and background noise is mitigated. In this technique, one can measure the delay of the inverters of stages (e.g., GUT I_1 (refer to Figure 2.5) of stages 2, 3, 4 and 5). However, the delay of inverter I_1 of first stage will be different from I_1 inverter of intermediate stage. This is because, the last stage is driving a long inter-connect while the slew at the first stage will be different as compared to intermediate stage. That is the reason even though we have 11-stage GDMCs in each RO, we have shown the results for 10 gates.

The minimum number of stages required for this type of delay measurement is two and five for non-inverting and inverting gates, respectively. Since the RO requires an odd number of stages, the number of equations formed using five-stage RO is $C(5, 5) + C(5, 3) = 11$ and the number of variables (e.g., gate delays to be measured) is 5. Hence, the delays can be cross-checked across many different measurements similar to Reference 27. For our test chip, measurements of 22 different gates each with six different configurations yield the delay values for the corresponding gates to be within 0.64 ps (refer to sub-section 2.5.8), indicating the robustness of the measurements.

The technique presented can be applied to any logic gate by replacing GUT I_1 of Figure 2.5. This enables the measurement of local and global variation of any logic gate in the standard cell library.

The circuit structure in Figure 2.8 was fabricated in silicon in a 65 nm process node. Careful attention was paid to power routing to ensure that any variations were due to process and not due to voltage drop in power grid. Hence, a uniform grid was laid out and also a ball grid array (BGA) package was chosen to ensure high-quality power distribution with minimal voltage drop across the cells in the structure.

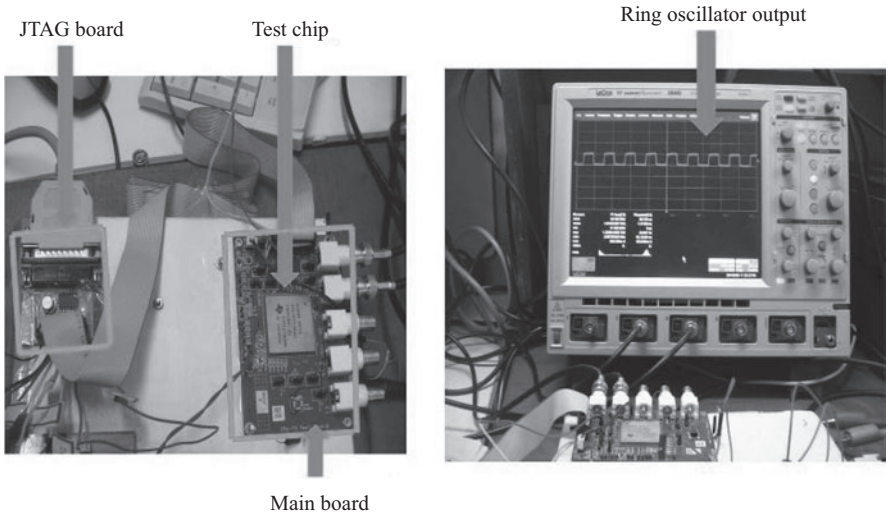


Figure 2.9 Experimental setup showing test chip with main board and oscilloscope output [28]. © 2009 IEEE

2.5.3 Measured results

Each logic gate's delay measurement requires period measurements of four and two different oscillator configurations for inverting and non-inverting cell, respectively, as described in (2.8) and (2.15). Many different sets of configuration words can yield the delay of the same gate in (2.15). Measurements of 11 different gates each with six different configurations yield the delay values for the same gate to be within 0.64 ps, indicating the robustness of the measurements. The experimental setup with the main board and RO output in oscilloscope is shown in Figure 2.9.

Figure 2.10 shows the measured individual delays of inverter I_1 (refer to Figure 2.5) (normalized to a fan-out one loaded inverter) in each of the 10 stages of the same ring from two different chips. The inverter-to-inverter delay in the ring, as a percentage of the mean delay varies by up to 26% for chip1 and 17.4% for chip2, indicating the effect of intra-chip local variations. There is no discernible pattern of variation for the delays within the same ring. Between the two chips, the variation pattern has some similarity, for stages 4–8, but is different for the rest indicating the randomness of these local variations.

Figures 2.11 and 2.12 show the measured delay spread for inverter I_1 among the 10 stages for all the nine different ROs. The cell sizes and layout of I_1 for all the 10 stages within each ring is the same. However, there are small changes in the layout, for each RO, in terms of poly-pitch spacing as well as diffusion widths. We observe local delay spreads among nominally identical inverters within each ring, from 18% to 28% across the nine rings, again indicating the significant impact of local variations. Figures 2.13 and 2.14 show the delay spread for the same rings from chip2.

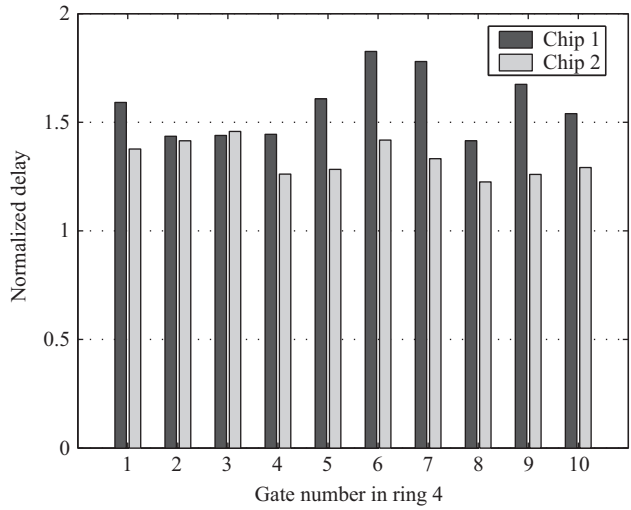


Figure 2.10 Measured delays of 10 nominally identical inverters in the same ring from different chips [28]. © 2009 IEEE

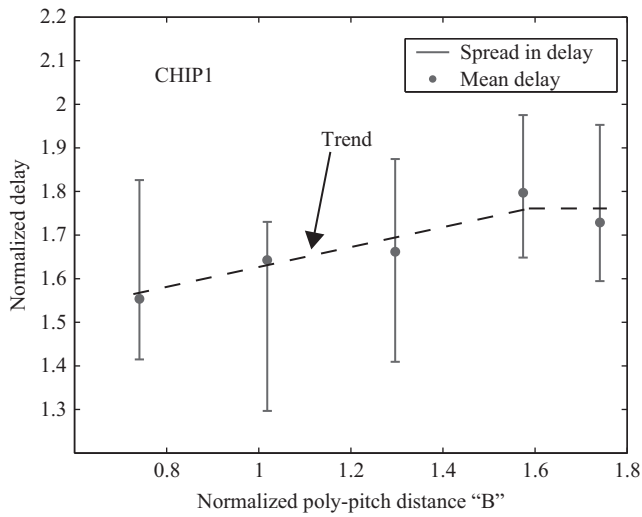


Figure 2.11 Measured mean gate delay and its spread over 10 nominally identical gates in each ring for five different rings for chip1 to study poly-pitch effect [28]. © 2009 IEEE

2.5.4 Poly-pitch effect

As mentioned earlier, five ROs are used to study the delay variation due to poly-pitch distance. Note that the length of transistor of gates in all the five ROs are same; however, poly-pitch parameter "*B*" as shown in Figure 2.15 is same in each ring

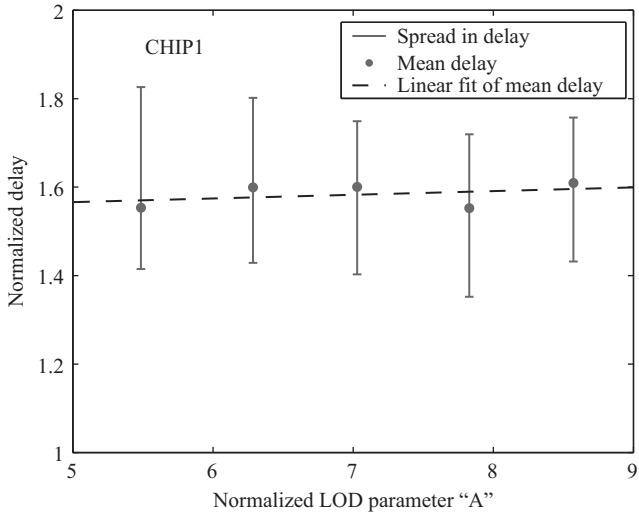


Figure 2.12 Measured mean gate delay and its spread over 10 nominally identical gates in each ring for five different rings for chip1 to study LOD effect [28]. © 2009 IEEE

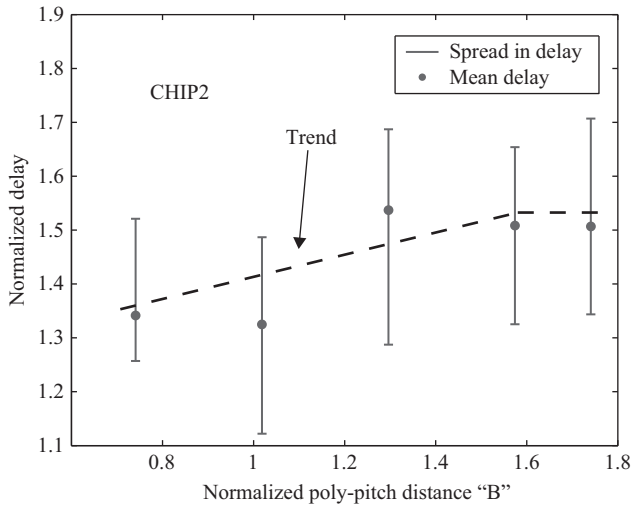


Figure 2.13 Measured mean gate delay and its spread over 10 nominally identical gates in each ring for five different rings for chip2 to study poly-pitch effect [28]. © 2009 IEEE

and varied among the rings. From Figures 2.11 and 2.13, it is observed that mean delay of the gate increases with increase in normalized poly-pitch distance from 0.74 to 1.57 and remains constant beyond 1.57 for both the chips. As observed in References 34 and 35, the frequency of the RO decreases with increase of poly-pitch

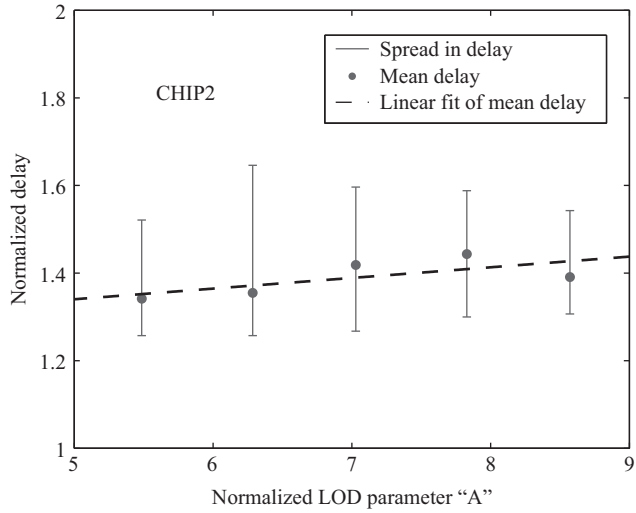


Figure 2.14 Measured mean gate delay and its spread over 10 nominally identical gates in each ring for five different rings for chip2 to study LOD effect [28]. © 2009 IEEE

or poly-silicon spacing and after certain distance frequency remains constant. This is because, the poly-silicon spacing impacts frequency or in our case individual gate delay due to photolithography. When the distance between poly pitch is increased, effective length of transistor in silicon increases and hence the mean delay of the gate increases. As observed, the delay variance of the gate delay does not depend on the poly-silicon spacing.

2.5.5 Length of diffusion effect

Five ROs are dedicated for studying delay variation due to change in active overhang or length of diffusion (LOD). Note that the length of transistor of gates in all the five ROs are same; however, LOD parameter "*A*" as shown in Figure 2.15 is same in each ring and varied among the rings. From Figures 2.12 and 2.14, it is observed that mean gate delay in these ROs does not depend upon change in LOD. The simulation results in Reference 36 show that the threshold voltage V_{th} of N-channel metal oxide semiconductor field effect transistor (NMOS) increases with decreasing LOD and saturates at higher LOD for same gate length. In case of P-channel metal oxide semiconductor field effect transistor (PMOS), V_{th} does not change with active overhang. Since in our experimentation, the normalized LOD lies between 5.48 and 8.57 which is too large to show up stress due to shallow trench isolation (STI), hence the mean delay of individual gate remains unchanged. It is important to mention that each point in Figures 2.11–2.14 is obtained from 10 delay numbers measured from 10 identical gates in each RO.

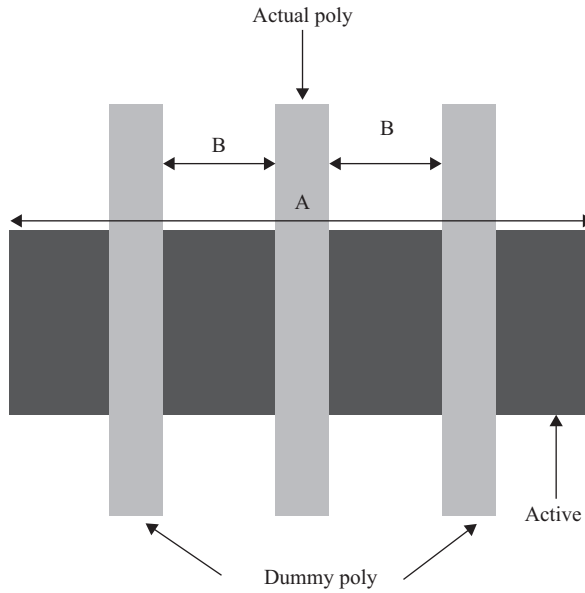


Figure 2.15 Modified transistor layout to study poly-pitch effect and length of diffusion (LOD) effect showing "A" as active overhang or LOD and "B" as poly-pitch distance [28]. © 2009 IEEE

2.5.6 Delay variation due to layout orientation

Figure 2.16 shows the delays of the 10 inverters in two ROs, one laid out vertically and the other horizontally for chip1, showing a delay spread of 19%. Note that in these two rings, the inverter I_1 has a different size, layout and loading compared to that of Figure 2.11. The pattern of local delay variation is different for the vertical and horizontal cells. Figure 2.17 shows the delays of the 10 identical inverters in horizontal and vertical ROs for chip2. Comparison of Figures 2.16 and 2.17 shows the gate numbers 1, 2, 3, 4, 8 and 10 have a pattern of similarity whereas the rest of the gates shows opposite pattern showing the randomness of delay variation.

2.5.7 Delay variation due to supply voltage

Figure 2.18 shows the delay variations of the 10 cells in a ring for two different supply voltages. We observe that the delay and the delay spread at 0.8 V are more than that for 1 V, but the pattern of variation is the same. As we decrease the supply voltage, the spread in delay increases from 17% at 1 V to 29% at 0.8 V, indicating that the main source of local variations is the process. Because, the impact of process variation is more at reduced gate overdrive (supply voltage minus threshold voltage).

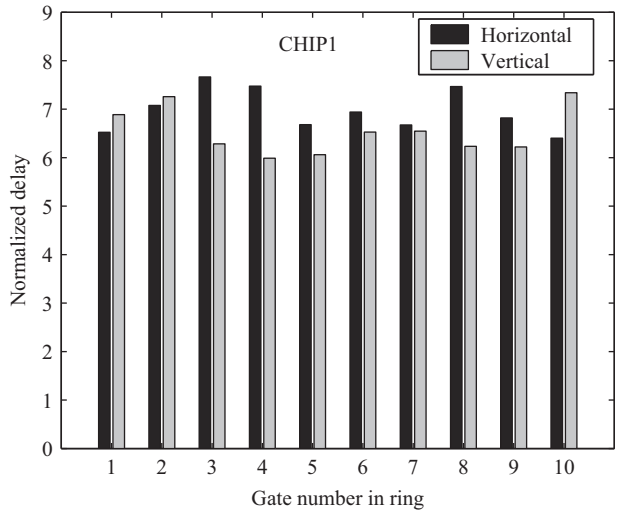


Figure 2.16 Measured delays of 10 nominally identical inverters in horizontal and vertical orientation RO of chip1 [28]. © 2009 IEEE

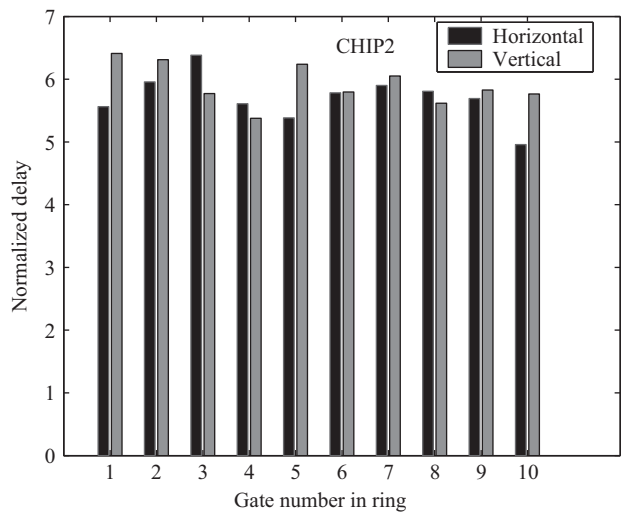


Figure 2.17 Measured delays of 10 nominally identical inverters in horizontal and vertical orientation RO of chip2 [28]. © 2009 IEEE

2.5.8 Measured accuracy of the delay measurement

As mentioned earlier, the same gate's delay can be measured in multiple ways, because the number of equations is more than the number of variables. Note that each equation corresponds to a specific configuration of the RO, while each variable corresponds to

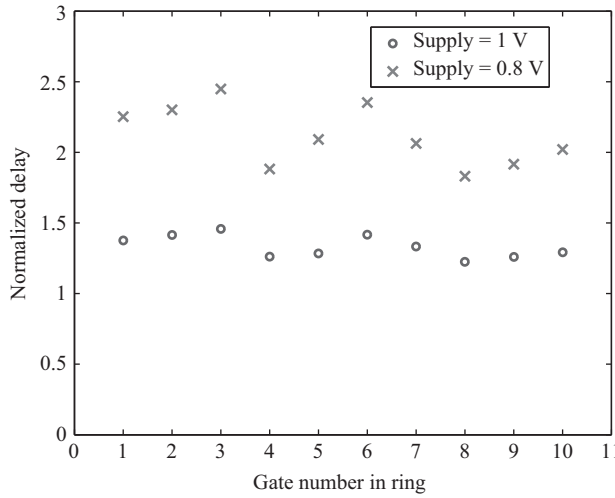


Figure 2.18 Measured gate delays in a ring for 1 V and 0.8 V supply [28]. © 2009 IEEE

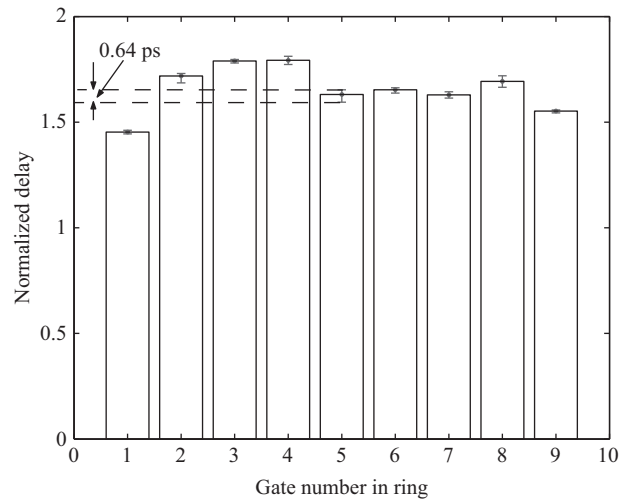


Figure 2.19 Error in delay measurement of different gates [28]. © 2009 IEEE

the delay of the GUT in each configurable stage. Figure 2.19 shows the delay of nine different gates, with each gate delay measured in six different ways. The maximum error in a gate's delay is found to be 0.64 ps and is quite small compared to measured delay of the gate.

We have also compared the measured delay results with statistical SPICE simulations. The mean delays are very close to that predicted by SPICE. While we do not have enough measured data points to infer about the distribution and correlations, we

Table 2.2 Comparison with other gate delay measurement techniques [28] © 2009 IEEE

Parameters	**[27]**	**[29]**	**[25]**	**[16, 20]**	**[28]**
Area	2 Inv	1 Inv** + 1 Mux***	Higher	1 Inv	5 Inv + 2 Mux
Technology	180 nm CMOS*	180 nm CMOS	180 nm CMOS	65/90 nm CMOS	65 nm CMOS
Accuracy	<1 ps	Not reported	About 1 ps	Not reported	<1 ps
Feasibility of self-validation	Yes	Yes	No	No	Yes
Unmodified cell	No	No	Yes	Yes	Yes
Inverting gate delay measurement	No	No	Yes	Yes	Yes
Non-inverting gate delay measurement	No	No	Yes	No	Yes
Robustness to PVT variation	Yes	Yes	No	No	Yes
Inter-die delay variation	No	No	Yes	Yes	Yes
Intra-die delay variation	Yes	Yes	Yes	No	Yes

* complementary metal-oxide semiconductor
** Inverter
*** Multiplexer

have found the measured data points to be largely within the distribution predicted by Monte Carlo SPICE simulations.

Our measurements confirm that local gate-to-gate variations are very significant in advanced process nodes and need to be accounted for in the models and design practices. The presented measurement technique is a simple yet powerful way to study these variations.

2.5.9 Comparison with other works

Table 2.2 compares our assessment of the different aspects of five delay measurement techniques. They have been designed in different technology nodes which make a true comparison difficult. The techniques of delay measurement in References 16, 20, 27, 28 and 29 are all RO-based designs. The delay measurement in Reference 25 is based on random sampling and comes the closest in terms of its ability to measure a single gate's delay. However, the mismatch in samplers needs to be mitigated by using larger sizes for the input devices in the sampler. This will increase the area overhead, and more importantly create loading problems for the GUT. Our technique has a lower area overhead and it provides some benefits in local variability characterization, since one needs to measure a large numbers of devices. Another benefit of our technique is that the measurement itself does not suffer because of process, voltage and temperature (PVT) variations as it relies on the computation

of difference of linear delay equations. Additionally, our approach provides a way for self-validation of the measured delay, as the delay can be measured in multiple different ways.

2.6 Measurement of rise and fall delays using standard RO

This section explains the relationship between the rise and fall gate delays of all the gates in the path of a RO and the positive/negative duty cycle of the RO signal, which enables the die-to-die rise/fall delay variability measurement using a standard RO test structure. For simplicity of explanation, a five-stage normal RO is considered as shown in Figure 2.20. T_{ON} and T_{OFF} are the positive or ON and negative or OFF part of the duty cycle of RO, respectively. In summary, T_{OFF} of the RO is equal to the sum of rise delay of odd stage inverters and the fall delay of even stage inverters. T_{ON} of the RO is equal to the sum of fall delay of odd stage inverters and the rise delay of even stage inverters. D_i^R and D_i^F are the rise delay and fall delay of ith stage gate in the RO, respectively. Hence, the positive/negative duty cycle of the RO signal is represented as follows:

$$\sum_{i=1,3,5} D_i^F + \sum_{j=2,4} D_i^R = T_{ON} \tag{2.16}$$

$$\sum_{i=1,3,5} D_i^R + \sum_{j=2,4} D_i^F = T_{OFF} \tag{2.17}$$

Assuming each gate in the RO has equal rise delay and fall delay inside a given chip due to die-to-die process variation. Let us define D^R and D^F as the rise delay and fall delay of each inverter in the RO, respectively. Then, positive and negative duty cycle of the RO signal can be represented in terms of rise delay D^R and fall delay D^F of the gates in the path of RO as follows:

$$3D^F + 2D^R = T_{ON} \tag{2.18}$$

$$2D^F + 3D^R = T_{OFF} \tag{2.19}$$

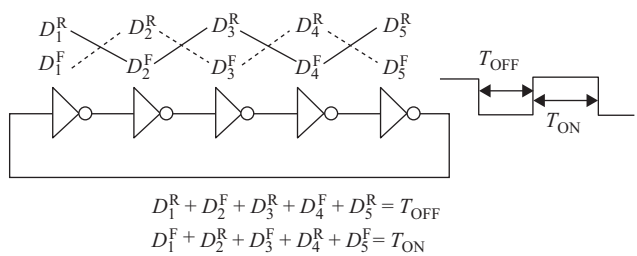


Figure 2.20 Relation between rise and fall delays of gates in the path of RO with the duty cycle of RO [37]. © 2014 IEEE

Solving (2.18) and (2.19), we get

$$D^{\mathrm{R}} = \frac{(3T_{\mathrm{OFF}} - 2T_{\mathrm{ON}})}{5} \tag{2.20}$$

$$D^{\mathrm{F}} = \frac{(3T_{\mathrm{ON}} - 2T_{\mathrm{OFF}})}{5} \tag{2.21}$$

Using (2.20) and (2.21) represents how the rise delay and fall delay of each inverter of the RO can be found out from the positive and negative duty cycle of the RO signal. The rise delay and fall delay calculations for N-stage RO can be generalized as follows:

$$D^{\mathrm{R}} = \frac{[((N+1)/2)T_{\mathrm{OFF}} - ((N-1)/2)T_{\mathrm{ON}}]}{N} \tag{2.22}$$

$$D^{\mathrm{F}} = \frac{[((N+1)/2)T_{\mathrm{ON}} - ((N-1)/2)T_{\mathrm{OFF}}]}{N} \tag{2.23}$$

Equations (2.22) and (2.23) are useful to find out die-to-die rise delay and fall delay variation of the inverting gates in the RO. However, the rise and fall delays of the non-inverting gate cannot be measured using this technique. To measure the within-die delay variability, a specialized RO structure is required as explained in the next section.

2.7 Rise and fall gate delay measurement using RRO

In this section, a novel measurement technique of within-die rise/fall delay variability of individual gate is presented. For simplicity of explanation, a three-stage RRO is used in case of non-inverting gate whereas a five-stage RRO is used in the case of inverting gate. However, the technique is valid for any number of stages of the RRO.

2.7.1 Gate delay measurement cell

The relation of rise delay and fall delay of the gates in the RO to the duty cycle of the RO is extended to a RRO which is shown in Figure 2.21. The RRO consists of a chain of back-to-back connected GDMCs which contains the GUT. The GUT can be inverting or non-inverting gate. The multiplexer Mux_1 in GDMC will determine when to keep the GUT in the RO path based on the select input. The detailed explanations regarding dummy GUT, multiplexer Mux_2 and inverter I_2 are available in References 38 and 28. However, the GDMC is slightly different from that in References 38 and 28 as the latter is non-inverting in nature. The inverting nature of the GDMC is useful while measuring the rise/fall delay of inverting gate. The rise/fall delay of an individual gate can be determined by taking the difference of duty cycle for two status vectors of the

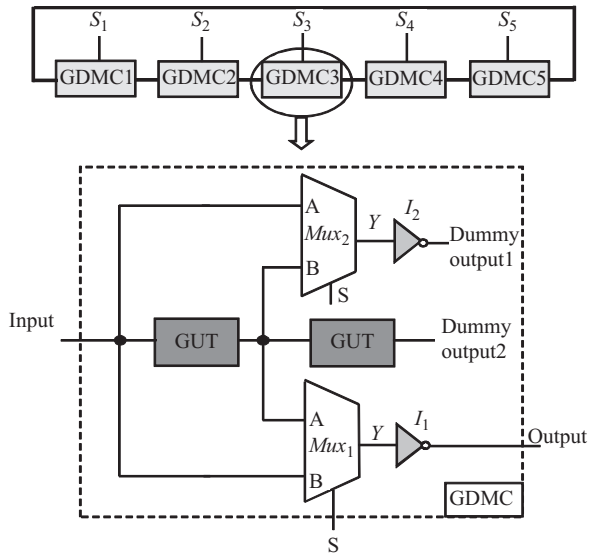


Figure 2.21 RRO consisting of GDMC [37]. © 2014 IEEE

RRO (i.e., one with and another without GUT in the path). In case of non-inverting gate, the rise/fall delay of the individual gate is calculated directly. However, in case of inverting gate, rise/fall delay of two gates is calculated. The accurate measurement of duty cycle is very important for accurate rise/fall delay measurement.

2.7.2 Rise and fall delays of non-inverting gate

For simplicity of explanation, a three-stage RRO (i.e., $K = 3$) is chosen as shown in Figure 2.22. In this case, all three GUTs are non-inverting gates. Table 2.3 depicts the notations and definitions of variables used to establish the relationship of the rise and fall delays of the gates in the path of RRO with the duty cycle of RRO.

Let S_i is the select input of multiplexer. When $S_i = 0$, the path from input A to output Y of the multiplexer Mux_1 of the ith stage GDMC is connected. When $S_i = 1$, the path from input B to output Y of the multiplexer Mux_1 of the ith stage GDMC is connected.

The status vector $S = \{S_i\}$ is defined as a 3-bit vector as three-stage RRO is considered. When status vector $S = \{101\}$, then the multiplexer input B to output Y of stages 1 and 3 are connected and the multiplexer input A to output Y of stage 2 is connected. The relationships of rise and fall delays of all the gates in the path of the RRO with the positive/negative duty cycle are as follows:

$$D_{m,1}^{\mathrm{B,F}} + D_{2}^{\mathrm{F}} + D_{m,2}^{\mathrm{A,R}} + D_{m,3}^{\mathrm{B,F}} = T_{\mathrm{ON}}^{1} \tag{2.24}$$

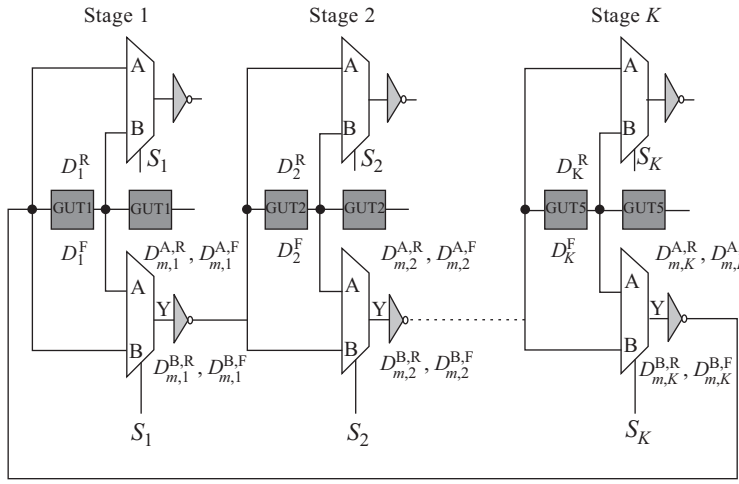


Figure 2.22 Measurement scheme of rise/fall gate delay using Kth-stage RRO. Assume $K = 3$ in case of non-inverting gates in the RRO and $K = 5$ in case of inverting gates in the RRO for simplicity of explanation [37]. © 2014 IEEE

Table 2.3 Notations and definitions of variables for rise and fall delays measurement [37] © 2014 IEEE

Notations	Definitions of variables
D_i^R and D_i^F	Rise and fall delays of ith stage GUT, respectively
$D_{m,i}^{A,R}$	Rise delay from input A of the multiplexer to the output of inverter I_1 of ith stage GDMC
$D_{m,i}^{A,F}$	Fall delay from input A of the multiplexer to the output of inverter I_1 of ith stage GDMC
$D_{m,i}^{B,R}$	Rise delay from input B of the multiplexer to the output of inverter I_1 of ith stage GDMC
$D_{m,i}^{B,F}$	Fall delay from input B of the multiplexer to the output of inverter I_1 of ith stage GDMC
S	Status vector of the RRO
T_{ON} and T_{OFF}	Positive and negative duty cycle of undivided RRO signal, respectively

and

$$D_{m,1}^{B,R} + D_2^R + D_{m,2}^{A,F} + D_{m,3}^{B,R} = T_{OFF}^1 \tag{2.25}$$

Similarly, when status vector $S = \{111\}$, the relationships of rise and fall delays of all the gates in the path of the RRO with the positive/negative duty cycle are as follows:

$$D_{m,1}^{B,F} + D_{m,2}^{B,R} + D_{m,3}^{B,F} = T_{ON}^2 \tag{2.26}$$

and

$$D_{m,1}^{\mathrm{B,R}} + D_{m,2}^{\mathrm{B,F}} + D_{m,3}^{\mathrm{B,R}} = T_{\mathrm{OFF}}^{2} \tag{2.27}$$

Solving (2.24) and (2.26) gives the fall delay of the non-inverting GUT,

$$D_{2}^{\mathrm{F}} = \left(T_{\mathrm{ON}}^{1} - T_{\mathrm{ON}}^{2}\right) - \left(D_{m,2}^{\mathrm{A,R}} - D_{m,2}^{\mathrm{B,R}}\right) \tag{2.28}$$

and solving (2.25) and (2.27) gives the rise delay of the non-inverting GUT,

$$D_{2}^{\mathrm{R}} = \left(T_{\mathrm{OFF}}^{1} - T_{\mathrm{OFF}}^{2}\right) - \left(D_{m,2}^{\mathrm{A,F}} - D_{m,2}^{\mathrm{B,F}}\right) \tag{2.29}$$

2.7.3 Rise and fall delays of inverting gate

Figure 2.22 shows a five-stage RRO (i.e., $K = 5$) to explain the measurement of rise/fall delay of inverting gate. In this case, all five GUTs are inverting gates. For status vector $S = \{10000 \text{ and } 11100\}$, four equations can be constituted relating the rise and fall delays of GUT, multiplexer and inverter to the duty cycle of the RO signal.

When status vector $S = \{10000\}$, the relationships of rise and fall delays of all the gates in the path of the RRO with the positive/negative duty cycle are as follows:

$$D_{m,1}^{\mathrm{B,F}} + \sum_{i=2}^{5} D_{i}^{\mathrm{R}} + \sum_{i=2}^{5} D_{m,i}^{\mathrm{A,F}} = T_{\mathrm{ON}}^{3} \tag{2.30}$$

and

$$D_{m,1}^{\mathrm{B,R}} + \sum_{i=2}^{5} D_{i}^{\mathrm{F}} + \sum_{i=2}^{5} D_{m,i}^{\mathrm{A,R}} = T_{\mathrm{OFF}}^{3} \tag{2.31}$$

When status vector S={11100}, the relationships of rise and fall delays of all the gates in the path of the RRO with the positive/negative duty cycle are as follows:

$$D_{m,2}^{\mathrm{B,R}} + \sum_{i=1,3} D_{m,i}^{\mathrm{B,F}} + \sum_{i=4}^{5} D_{i}^{\mathrm{R}} + \sum_{i=4}^{5} D_{m,i}^{\mathrm{A,F}} = T_{\mathrm{ON}}^{4} \tag{2.32}$$

and

$$D_{m,2}^{\mathrm{B,F}} + \sum_{i=1,3} D_{m,i}^{\mathrm{B,R}} + \sum_{i=4}^{5} D_{i}^{\mathrm{F}} + \sum_{i=4}^{5} D_{m,i}^{\mathrm{A,R}} = T_{\mathrm{OFF}}^{4} \tag{2.33}$$

Solving (2.30) and (2.32) gives sum of the rise delay of the two inverting GUTs,

$$\sum_{i=2}^{3} D_{i}^{\mathrm{R}} = \left(T_{\mathrm{ON}}^{3} - T_{\mathrm{ON}}^{4}\right) - \left[\sum_{i=2}^{3} D_{m,i}^{\mathrm{A,F}} - \left(D_{m,2}^{\mathrm{B,R}} + D_{m,3}^{\mathrm{B,F}}\right)\right] \tag{2.34}$$

and solving (2.31) and (2.33) gives sum of the fall delay of two inverting GUTs,

$$\sum_{i=2}^{3} D_i^{\mathrm{F}} = \left(T_{\mathrm{OFF}}^{3} - T_{\mathrm{OFF}}^{4}\right) - \left[\sum_{i=2}^{3} D_{m,i}^{\mathrm{A,R}} - \left(D_{m,2}^{\mathrm{B,F}} + D_{m,3}^{\mathrm{B,R}}\right)\right] \tag{2.35}$$

The first term in the right-hand side of (2.28), (2.29), (2.34) and (2.35) is measured from the duty cycle of the RRO. The second term in the right-hand side of (2.28), (2.29), (2.34) and (2.35) is calculated from simulation and subtracted from the first term of (2.28), (2.29), (2.34) and (2.35) in order to evaluate the respective rise/fall gate delay. The second term in the right-hand side of (2.28), (2.29), (2.34) and (2.35) can be minimized by proper sizing of A to Y and B to Y path of the multiplexer. In case of inverting gate, the sum of rise/fall delay of two gates can be measured. The rise/fall delay of single gate can be estimated statistically. Let the mean and standard deviation of sum of rise/fall delay of two gates are μ_m and σ_m, respectively. It is assumed that the rise/fall delay of single gate follows uncorrelated Gaussian distribution with mean μ_{GUT} and standard deviation σ_{GUT}. Then, the mean μ_{GUT} and standard deviation σ_{GUT} of rise/fall delay of single gate are represented as $\mu_m/2$ and $\sigma_m/\sqrt{2}$, respectively.

2.8 Test chip and measurement results

2.8.1 Measurement of duty cycle

This section explains how the duty cycle of a very high-frequency RRO signal can be efficiently measured using the sub-sampling principle [39]. In general, the frequency of the undivided RO is very high. It is difficult to bring the high-frequency signal out of the chip. A sub-sampling unit is implemented inside the chip to slow down the signal without losing the duty cycle information of the undivided RRO signal. The conventional method of converting high-frequency signal to low-frequency signal using divider is not suitable in this measurement as the positive/negative duty cycle becomes equal at the output of a divider.

The sub-sampling unit is implemented using an edge-triggered flip-flop with the undivided RRO signal connected to its data input and the output of the phase locked loop (PLL) is connected to the clock input as shown in Figure 2.23. Let T is the period of the undivided high-frequency RRO signal. The clock of period $(T \pm \Delta T)$ is essential for the proper operation of sub-sampling and is generated using an on-chip PLL. The undivided RRO signal is first divided down by a programmable divide-by-$(N-1)$ circuit. The period of the output signal of the programmable divide-by-$(N-1)$ is $T(N-1)$, which is fed as the input reference clock signal to the PLL. The PLL multiples the input reference clock signal by N which generates clock of period $(T(N-1)/N)$. In this scheme, the value of ΔT is T/N. The output of the flip-flop is the low-frequency sub-sampled signal whose period is $(T/\Delta T)(T-\Delta T)$. The duty cycle of the sub-sampled RRO signal contains the information of the duty cycle of undivided RRO signal. The duty cycle of the undivided RRO signal can be

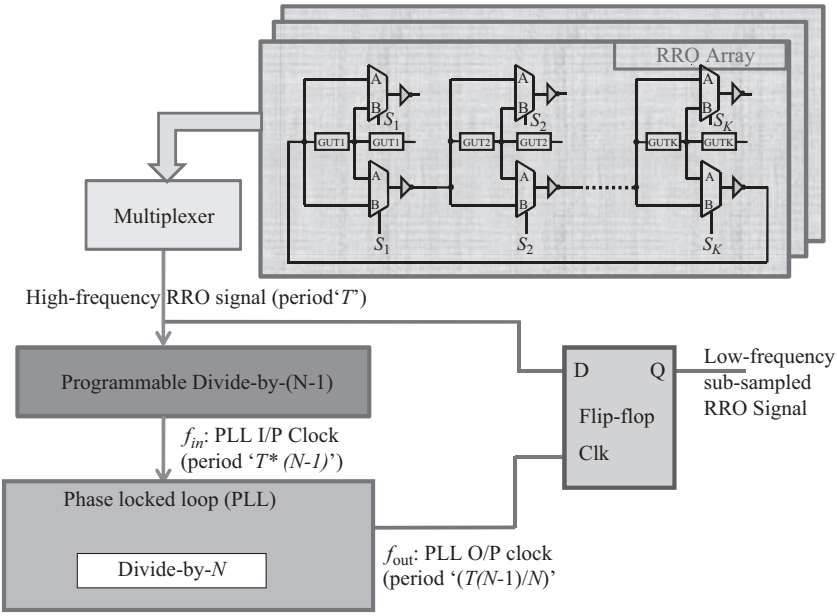


Figure 2.23 Sub-sampling unit with PLL for generating ($T \pm \Delta T$) clock [37]. © 2014 IEEE

calculated by dividing the duty cycle of the sub-sampled signal by $(N - 1)$. A conceptual timing diagram of the sub-sampling of a high-frequency RRO signal is shown in Figure 2.24. The positive/negative duty cycle of the high-frequency RRO signal is intentionally made unequal in Figure 2.24 to show the effect of sub-sampling on it. It shows how a high-frequency RRO signal can be converted to a low-frequency RRO signal using the sub-sampling.

One can avoid the use of a PLL if a high-frequency Input/Output (I/O) is available such that (1) provide a sub-sampling clock from an external clock generator or (2) bring the high-frequency RRO signal outside the chip for direct measurement of the waveform. In those cases, the PLL is not required which greatly reduces the complexity of the on-chip circuit.

2.9 Measured results

The test chip is fabricated in an industrial 65 nm process to measure the rise/fall delay of individual gate which is useful to study the advanced aging effect such as NBTI and Positive Bias Temperature Instability (PBTI) in the nanoscale technology node. The chip micrograph is shown in Figure 2.25 with dimension of the RRO block and PLL. Figure 2.26 shows the sub-sampled RRO signal, divided RRO signal and PLL lock signal in the oscilloscope screen. The divided RRO signal is the output of the programmable divider in Figure 2.23. In case of the divided RRO signal, the positive

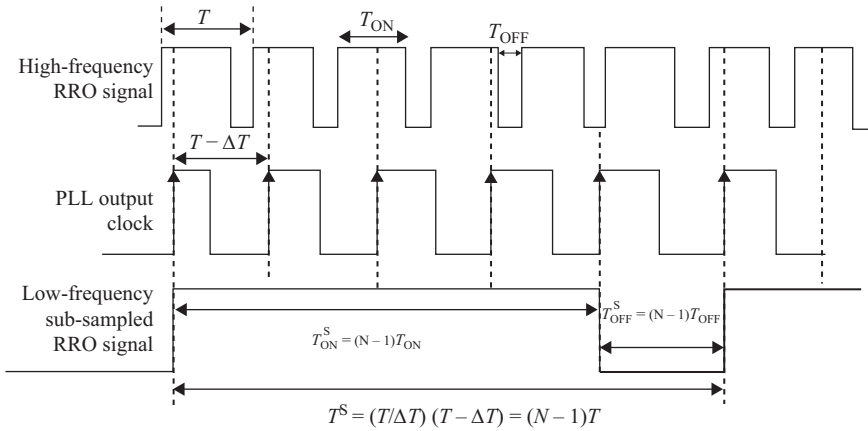


Figure 2.24 Conceptual timing diagram showing the high-frequency RRO signal, PLL output clock and low-frequency sub-sampled RRO signal with time-expanded duty cycle. [37]. © 2014 IEEE

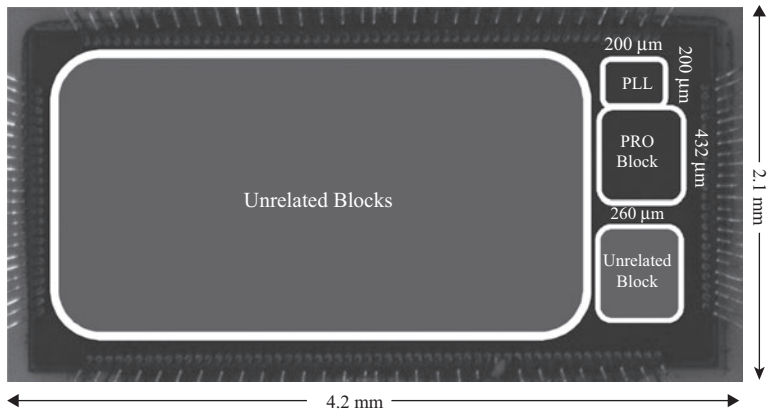


Figure 2.25 Showing the chip micrograph with the RRO block and PLL [37]. © 2014 IEEE

and negative duty cycle of the signal is equal and it does not carry the duty cycle information of undivided RRO signal. In case of the sub-sampled RRO signal, the positive and negative duty cycle of the signal is unequal as it preserves the duty cycle information of undivided RRO signal. Hence, the duty cycle of the sub-sampled RRO signal is useful to extract the rise and fall delays of individual logic gate. Some measured results from the test chip are presented below.

2.9.1 Impact of body-bias

The impact of body-bias voltage on the within-die variation of rise and fall delays is investigated. Figure 2.27(a) shows the rise and fall delays variability of buffer across

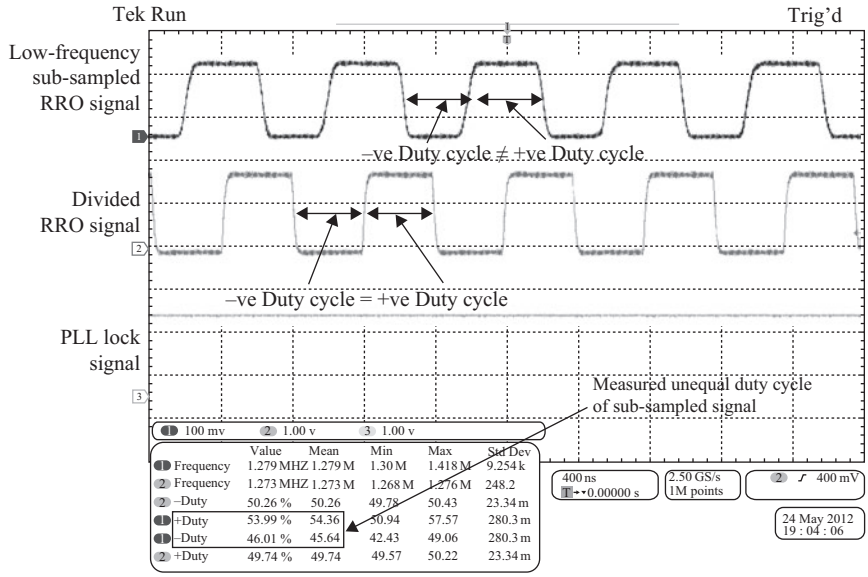

Tek Run
Trig'd
Low-frequency sub-sampled RRO signal
–ve Duty cycle ≠ +ve Duty cycle
Divided RRO signal
–ve Duty cycle = +ve Duty cycle
PLL lock signal
Measured unequal duty cycle of sub-sampled signal
100 mv
1.00 v
1.00 v
Value Mean Min Max Std Dev
Frequency 1.279 MHZ 1.279 M 1.30 M 1.418 M 9.254 k
Frequency 1.273 MHZ 1.273 M 1.268 M 1.276 M 248.2
–Duty 50.26 % 50.26 49.78 50.43 23.34 m
+Duty 53.99 % 54.36 50.94 57.57 280.3 m
–Duty 46.01 % 45.64 42.43 49.06 280.3 m
+Duty 49.74 % 49.74 49.57 50.22 23.34 m
400 ns
0.00000 s
2.50 GS/s
1M points
400 mV
24 May 2012
19 : 04 : 06

Figure 2.26 Oscilloscope waveform showing the low-frequency sub-sampled RRO signal, divided RRO signal and PLL lock signal [37]. © 2014 IEEE

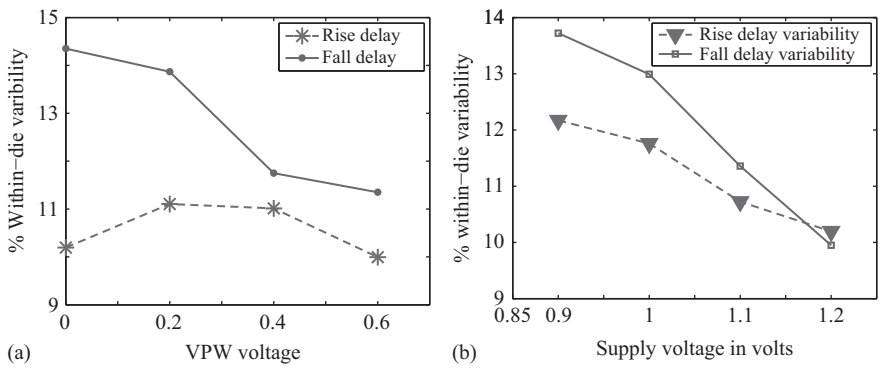


Figure 2.27 Measured buffer rise and fall delays across (a) P-well body bias (VPW) and (b) supply voltage [37]. © 2014 IEEE

P-well body bias (VPW). It is found that the fall delay variability decreases with the increase in the VPW, whereas the rise delay variability does not change much with the variation in VPW as shown in Figure 2.27(a). The distributions of rise and fall delays at VPW of 0.2 V are shown in Figures 2.28(a) and 2.28(b), respectively. Let σ and μ are the standard deviation and mean of the delay distribution, respectively. (σ/μ) is the measure of the within-die variability. It is found that the variabilities of the rise and fall delays (σ/μ) are 11% and 13.87%, respectively, at VPW of 0.2 V.

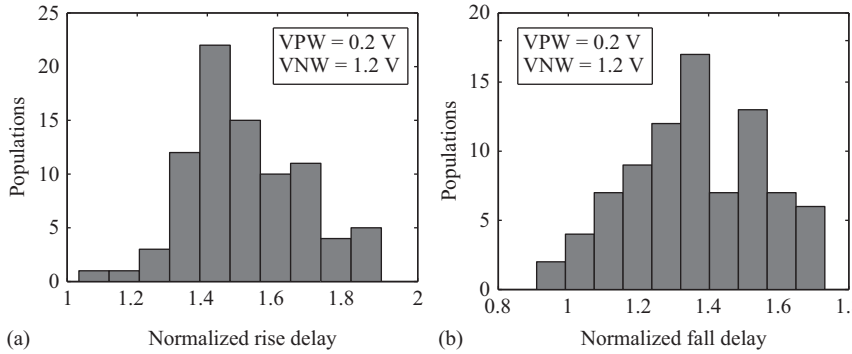


Figure 2.28 Measured buffer rise and fall delays distribution at P-well bias = 0.2 V and N-well bias = 1.2 V (a) rise delay and (b) fall delay [37]. © 2014 IEEE

2.9.2 Impact of supply voltage

The impact of supply voltage on the within-die variation of rise and fall gate delays is studied. Figure 2.27(b) shows the measured variabilities of rise and fall delays of buffer across supply voltages. It is found that the fall delay variability (σ/μ) is more than the rise delay variability at low supply voltage of 0.9 V. The variabilities of rise and fall delays are equal at nominal supply voltage such as 1.2 V. The variabilities of rise and fall gate delays increase with the decrease of supply voltage which indicates that the main source of within-die variations is due to the fabrication process. This is evident that the impact of process variation is predominant at reduced gate overdrive (supply voltage minus threshold voltage). The variabilities of the rise and fall delays are 12% and 13.7%, respectively, at supply voltage of 0.9 V. At nominal supply voltage of 1.2 V, the variabilities of the rise and fall delays are 10.2% and 9.95%, respectively.

2.9.3 Measurement accuracy

It is important to quantify the measurement accuracy of the rise and fall delays measurement. The rise and fall delays of the same gate (i.e., buffer) are measured at two different values of N, where N is the frequency multiplication factor of the PLL. Figure 2.29 shows the measured accuracy of eight GUTs in the RRO for two different values of N such as 85 and 86. It is found that the accuracy of rise delay for two different values of N is less than 1 ps as shown in Figure 2.29(a). The accuracy of fall delay for two different values of N is less than 1.5 ps as shown in Figure 2.29(b). It confirms the accuracy of the delay measurement scheme.

2.9.4 Comparison with the existing techniques

Table 2.4 compares the various aspects of four delay measurement techniques [25, 30, 31]. The analog measurement techniques such as random sampling [25] require sample and hold circuit at the input and output of GUT which is susceptible to process

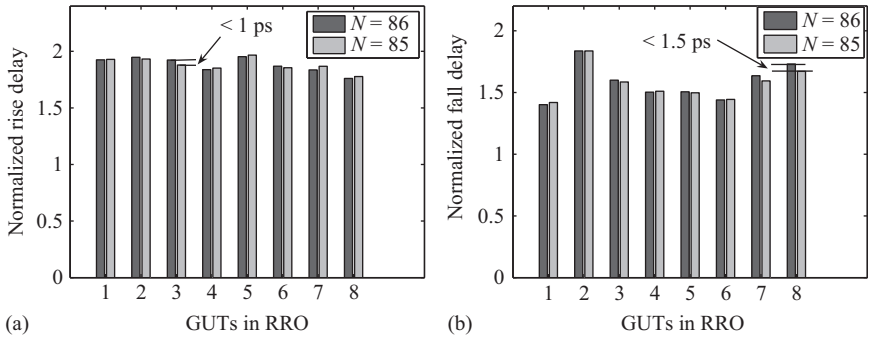


Figure 2.29 Accuracy of rise and fall buffer delay measurement using sub-sampling $N = 86$ and $N = 85$ (a) rise delay and (b) fall delay [37]. © 2014 IEEE

Table 2.4 Comparison of various rise and fall logic gate delay measurement techniques [37] © 2014 IEEE

	[25]	**[31]**	**[30]**	**[37]**
Principle	Random sampling	On-chip sampling scope	Pulse shrinking	RRO's duty cycle
Process	180 nm	65 nm	65 nm	65 nm
Digital/analog	Analog	Analog	Digital	Digital
Unmodified cell	Yes	Yes	No	Yes
Robustness to PVT variation	No	No	Yes	Yes

variation. In the on-chip sampling oscilloscope technique [31], each GUT requires a sampling head which is susceptible to process variation. Our proposed technique does not suffer from PVT variation as it relies on the computation of difference of linear delay equations. The pulse shrinking technique [30] requires resizing of the NMOS or PMOS network inside the GUT for the rise and fall delays measurement. However, the proposed technique can measure the rise and fall delays of the gate in its unmodified form (i.e., no resizing of transistors) in the standard cell library. As explained in Section 2.4, PLL is not mandatory requirement of this type of measurement. The essential part of the proposed circuit is a RRO that consists of digital gates only, without any analog components, which makes our circuit suitable to be implemented in a cell-based digital design environment.

2.10 Summary and conclusions

In this chapter, we have described a circuit technique to measure the average delay and rise/fall delay of individual gates of a standard cell library using RROs in silicon.

The average gate delay is measured using the frequency of the RROs and the rise/fall gate delay is measured using the duty cycle of the RROs. The high-frequency RRO signal is sub-sampled to generate the low-frequency RRO signal preserving the duty cycle information which is required for rise and fall delays measurements.

The easy on-chip implementation and sensitivity to process parameters of the RROs make it suitable for on-chip local variation measurement. Besides local variability, impact of local supply, temperature and neighborhood can also be studied. The inherent difference nature of delay measurement makes it immune to the error introduced due to systematic effects and background noise. The results from a 65 nm test chip show the efficacy of measurement to within 1 ps accuracy. Delay measurements of different, nominally identical, inverters in close physical proximity show variations of up to 28% indicating the large impact of local variations. The proposed technique is quite suitable for early process characterization, monitoring mature process in manufacturing, correlating model-to-hardware and studying local variation and neighborhood effects.

References

[1] M. Orshansky, L. Milor, and C. Hu, "Characterization of spatial intrafield gate CD variability, its impact on circuit performance, and spatial mask-level correction," *IEEE Transactions on Semiconductor Manufacturing*, vol. 17, no. 1, pp. 2–11, Feb. 2004.

[2] D. Blaauw, K. Chopra, A Srivastava, and L. Scheffer, "Statistical timing analysis: From basic principles to state of the art," *IEEE Transaction CAD*, vol. 27, no. 4, pp. 589–607, Apr. 2008.

[3] H. Chang and S.S. Sapatnekar, "Statistical timing analysis under spatial correlations," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 24, no. 9, pp. 1467–1482, Sep. 2005.

[4] K. Kuhn, Chris Kenyon, Avner Kornfeld, *et al.*, "Managing process variation in Intel's 45 nm CMOS technology," *Intel Technology Journal*, vol. 12, no. 2, pp. 93–109, 2008.

[5] K.J. Kuhn, Martin D. Giles, David Becher, *et al.*, "Process technology variation," *IEEE Transactions on Electron Devices*, vol. 58, no. 8, pp. 2197–2208, Aug. 2011.

[6] T. Mizuno, J.-I. Okamura, and A. Toriumi, "Experimental study of threshold voltage fluctuation due to statistical variation of channel dopant number in MOSFETs," *IEEE Transactions on Electron Devices*, vol. 41, no. 11, pp. 2216–2221, Nov. 1994.

[7] M.J. Pelgrom, C.J. Duinmaijer, and A.P.G. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1440, 1989.

[8] S. Xiong, J. Bokor, "Study of gate line edge roughness effects in 50 nm bulk MOSFET devices," *Proceedings of SPIE*, vol. 4689, pp. 733–741, 2002.

[9] A. Asenov, S. Kaya, and A.R. Brown, "Intrinsic parameter fluctuations in decananometer MOSFETs introduced by gate line edge roughness," *IEEE Transactions on Electron Devices*, vol. 50, no. 5, pp. 1254–1260, 2003.

[10] A. Asenov, S. Kaya, and J.H. Davies, "Intrinsic threshold voltage fluctuations in decanano MOSFETs due to local oxide thickness variations," *IEEE Transactions on Electron Devices*, vol. 49, no. 1, pp. 112–119, Jan. 2002.

[11] F. Klass, A. Jain, G. Hess, and B. Park, "An all-digital on-chip process-control monitor for process-variability measurements," in *Proceedings of IEEE ISSCC*, San Francisco, CA, USA, pp. 408–409, 2008.

[12] R. Rao, K.A. Jenkins, and J.-J. Kim, "A completely digital on-chip circuit for local-random-variability measurement," in *Proceedings of IEEE ISSCC*, San Francisco, CA, USA, pp. 412–413, 2008.

[13] S. Mukhopadhyay, K. Kim, K.A. Jenkins, C.-T. Chuang, and K. Roy, "Statistical characterization and on-chip measurement methods for local random variability of a process using sense-amplifier-based test structure," in *Proceedings of IEEE ISSCC*, San Francisco, CA, USA, pp. 400–401, 2007.

[14] E. Karl, P. Singh, D. Blaauw, and D. Sylvester, "Compact in-situ sensors for monitoring negative-bias-temperature-instability effect and oxide degradation," in *Proceedings of IEEE ISSCC*, San Francisco, CA, USA, pp. 410–411, 2008.

[15] N. Drego, A. Chandrakasan, and D. Boning, "A test-structure to efficiently study threshold-voltage variation in large MOSFET arrays," in *Proceedings of IEEE International Symposium on Quality Electronic Design*, San Jose, CA, USA, 2007.

[16] M. Bhushan, A. Gattiker, M.B. Ketchen, and K.K. Das, "Ring oscillators for CMOS process tuning and variability control," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 1, pp. 10–18, Feb. 2006.

[17] J. Panganiban, "A ring oscillator based variation test chip," M.Eng. Thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, May 2002.

[18] M. Ketchen and M. Bhushan, "Product-representative "at speed" test structures for CMOS characterization," *IBM Journal Research and Development*, vol. 50, no. 4/5, pp. 451–468, Jul./Sep. 2006.

[19] H. Onodera, "Variability: Modeling and its impact on design," *IEICE Transactions on Electronics*, vol. E89-C, no. 3, pp. 342–348, 2006.

[20] H. Masuda, S. Ohkawa, A. Kurokawa, and M. Aoki, "Challenge: Variability characterization and modeling for 65- to 90-nm processes," in *Proceedings of IEEE Conference on Custom Integrated Circuits*, San Jose, CA, USA, pp. 908–913, 2005.

[21] S. Ohkawa, M. Aoki, and H. Masuda, "Analysis and characterization of device variations in an LSI chip using an integrated device matrix array," *IEEE Transactions on Semiconductor Manufacturing*, vol. 17, no. 2, pp. 155–165, May 2004.

[22] N. Abaskharoun and G.W. Roberts, "Circuits for on-chip sub-nanosecond signal capture and characterization," in *Proceedings of IEEE Conference on Custom Integrated Circuits*, San Diego, CA, USA, pp. 251–254, 2001.

[23] P. Sanda, D. Knebel, J. Kash, H. Casal, J. Tsang, E. Seewann, and M. Papermaster, "Picosecond imaging circuit analysis of the power3 clock distribution," in *Proceedings of IEEE ISSCC*, San Francisco, CA, USA, pp. 372–373, 1999.

[24] D. Dajee, N. Goldblatt, T. Lundquist, S. Kasapi, and K. Wilsher, "Practical, non-invasive optical probing for flip-chip devices," in *Proceedings of IEEE International Test Conference*, Baltimore, MD, pp. 433–442, 2001.

[25] S. Maggioni, A. Veggetti, A. Bogliolo, and L. Croce, "Random sampling for on-chip characterization of standard cell propagation delay," in *Proceedings of IEEE International Symposium on Quality Electronic Design*, San Jose, CA, USA, 2003.

[26] K. Okada, K. Yamaoka, and H. Onodera, "A statistical gate-delay model considering intra-gate variability," in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, San Jose, CA, USA, pp. 908–913, 2003.

[27] A. Bassi, A. Veggetti, L. Croce, and A. Bogliolo, "Measuring the effects of process variations on circuit performance by means of digitally-controllable ring oscillators," in *Proceedings of IEEE International Conference on Microelectronic Test Structures*, Monterey, USA, pp. 214–217, 17–20 March 2003.

[28] B.P. Das, B. Amrutur, H.S. Jamadagni, N.V. Arvind, and V. Visvanathan, "Within-die gate delay variability measurement using reconfigurable ring oscillator," *IEEE TSM*, vol. 22, no. 2, pp. 256–267, May 2009.

[29] B. Zhou and A. Khouas, "Measurement of delay mismatch due to process variations by means of modified ring oscillators," in *Proceedings of IEEE International Symposium on Circuits and Systems*, Kobe, Japan, pp. 5246–5249, 2005.

[30] T. Iizuka, J. Jeong, T. Nakura, M. Ikeda, and K. Asada, "All-digital on-chip monitor for PMOS and NMOS process variability measurement utilizing buffer ring with pulse counter," in *Proceedings of ESSCIRC*, Seville, pp. 182–185, 2010.

[31] X. Zhang, K. Ishida, M. Takamiya, and T. Sakurai, "An on-chip characterizing system for within-die delay variation measurement of individual standard cells in 65-nm CMOS," in *Proceedings of IEEE ASP-DAC*, Yokohama, Japan, pp. 109–110, 2011.

[32] B.P. Das, B. Amrutur, H.S. Jamadagni, N.V. Arvind, and V. Visvanathan, "Within-die gate delay variability measurement using reconfigurable ring oscillator," in *Proceedings of IEEE Conference on Custom Integrated Circuits*, San Jose, CA, USA, pp. 133–136, 2008.

[33] B.P. Das, "Random local delay variability: On-chip measurement and modeling," Ph.D. Thesis, IISc, Bangalore, India, 2009.

[34] D. Boning, J. Panganiban, K. Gonzalez-Valentin, *et al.*, "Test structures for delay variability," in *Proceedings of Eighth ACM/IEEE International Workshop TAU*, Monterey, CA, USA, pp. 109, 2002.

[35] S.R. Nassif, D.S. Boning, and N. Hakim, "The care and feeding of your statistical static timer," in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, San Jose, CA, USA, pp. 138–139, 2004.

[36] M. Miyamoto, H. Ohta, Y. Kumagai, *et al.*, "Impact of reducing STI-induced stress on layout dependence of MOSFET characteristics," *IEEE Transactions on Electron Devices*, vol. 51, no. 3, pp. 440–443, Mar. 2004.

[37] B.P. Das and H. Onodera, "On-chip measurement of rise/fall gate delay using reconfigurable ring oscillator," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 3, pp. 183–187, Mar. 2014.

[38] B. Amrutur and B.P. Das, "Gate delay measurement circuit and method of determining a delay of a logic gate," US Patent No. 8,224,604 B1 and date of patent July 17, 2012.

[39] B. Amrutur, P.K. Das, and R. Vasudevamurthy, "0.84 ps resolution clock skew measurement via sub-sampling," *IEEE TVLSI*, vol. 19, no. 12, pp. 2267–2275, Dec. 2011.

Chapter 3

Nanoscale FinFET devices for PVT-aware SRAM

G.K. Sharma[1]*, Manisha Pattanaik*[1]
and Nandakishor Yadav[1]

This chapter describes nanoscale FinFET devices and their application in SRAM design. It also discusses variability of nanoscale integrated circuits (ICs) and introduces variability-aware memory design. In the previous two chapters, process variations were discussed for analog and digital ICs. However, this chapter focuses on futuristic memory design. A comprehensive variability including process, voltage and temperature (PVT) variations has been discussed for future SRAM design. After analysing the results of PVT-aware designs, it is found that sensitivity-driven IG-FinFET-based SRAM is the most suitable technique for reliable and high-density memories. The design of SRAM using a post-CMOS device, namely FinFET widely adopted in semiconductor industry has been specifically elaborated.

3.1 Introduction

The high-density reliable static random access memory (SRAM) is the bottleneck of current and future generation high-performance multicore embedded systems. So, scaling of memory density must continue as it occupies a large fraction of many state-of-the-art designs. The continued metal oxide semiconductor (MOS) device scaling results in reduced performance and stability of SRAM cells due to PVT variations [1]. The increased effect of voltage variation is due to the scaling of the supply voltage with respect to device dimension in order to achieve constant electric field. The scaled devices further increase the chip density by increasing the effect of temperature variations due to local hot spots and self-heating. Thus, the increased parameter variation poses various challenges in nanoscale SRAM cell. The performance, power, and reliability are the major issues of nanoscale complementary MOS (CMOS)-based SRAMs due to reduced ON-current and increased leakage current. Further, bulk

[1]ABV – Indian Institute of Information Technology and Management, Gwalior 474 015, Madhya Pradesh, India

CMOS scaling beyond 45nm technology causes large PVT variations due to high-channel doping to control the device effectively. Over the years, silicon-on-insulator (SOI) technology has been developed to enable higher system performance, more cache memory, better power management, leading to cheaper packaging solutions. In nanometre regime, SOI FinFET is emerged as an alternative device that addresses major unresolved issues of scaled MOSFETs. This device has overcome scaling issues by limiting drain induced barrier lowering (DIBL) and other short-channel effects for gate length below 45nm technology and achieves the performance as good as classical CMOS device [2]. The reduced short-channel effects enable low body doping, and therefore, the lightly doped channel of the FinFET improves its resistance to process variations. However, variations within most sensitive parameters such as device gate length, fin width, fin height and gate-oxide thickness may have large impact on the performance and reliability of scaled FinFET-based SRAM cells. This variation causes large increase in leakage power, thus results in larger temperature variations and degrades the thermal problems. Therefore, PVT-aware FinFET SRAM design has become an important area of concern for academia and industry in order to significantly improve the performance and reliability of system-on-chip (SOC).

This chapter presents the design and characterization of PVT-aware nanoscale FinFET-based SRAM. In subsequent sections, various nanoscale FinFET devices and FinFET-based SRAM topologies are discussed. In addition, FinFET-based SRAM design challenges, PVT-aware design techniques and their implementations are briefly presented.

3.2 Nanoscale FinFET devices

Nanoscale multi-gate devices have been gaining significant interest as alternative devices over the conventional MOSFETs. This is simply due to their key features such as high control over the channel, smaller subthreshold leakage and reduced susceptibility to process variations [3]. Double-gate MOSFET (DGMOS) is one such multi-gate device that addresses the short-channel effects, and FinFET is its first practical implementation [4].

Figure 3.1 shows the basic DGMOS structure. The DGMOS device is the sandwich of a fully depleted SOI device between two gate electrodes connected together. This provides an improved gate control over the conducting channel and depletion region. In the fully depleted SOI device, most of the electric field lines propagate through the buried oxide (BOX) before reaching the channel region. Thus, it reduces the influence of drain electric field and the short-channel effects [5]. Two inversion regions are created at the surface of the channel by applying the gate voltage. This doubles the ON-current (I_{ON}) but limits the OFF-current (I_{OFF}) because the front- and back-gate electric field reduces the leakage current between source and drain.

Frank *et al.* explored the crucial scaling of the DGMOS device using device modelling and presented the Monte-Carlo (MC) simulation results [6]. According to this model, the ideal perfect device is a DGMOS at gate length of 30 nm, oxide thickness (t_{ox}) of 3 nm, and a silicon film thickness (t_{Si}) of 5–20 nm. The practical

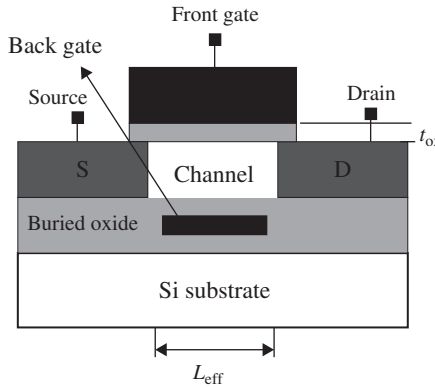


Figure 3.1 DGMOS structure

DGMOS is the fully depleted lean-channel transistor (DELTA) [7] with a thin fin and therefore known as DGFET or DG-FinFET. The term FinFET is first introduced by the researchers of University of California, Berkeley [8,9] for a nonplanar, double-gate transistor built on an SOI substrate. However, the planar DGFETs are found difficult to fabricate using the conventional CMOS technology [10]. This is because of the problems in aligning the top and buried bottom gate with a low-resistance gate contact. Moreover, the FinFET device can be easily fabricated by creating a thin fin vertically. This fin can be built within body (bulk) of the silicon or over the insulating layer. In general, the FinFET devices are classified into two types: bulk FinFET and SOI FinFET. Both kinds of FinFET devices are described in the following subsections.

3.2.1 Bulk FinFET

In the bulk FinFET, fins are etched on a bulk silicon wafer using the plasma etching and trimming via oxidation steps. Two sides of the fin are isolated using field oxide to avoid inversion between the fins [11]. Figure 3.2 shows the device structure of the bulk FinFET having a fin body connected to the substrate directly. The n+ and p+ poly-silicon gates are created on the n− and p-channel bulk FinFETs, respectively. The gate-oxide thickness of the bulk FinFET is symmetric to the top, front and back surfaces. The field oxide thickness (T_{FOX}) and the silicon nitride (SiN) thickness are mostly thicker than the gate oxide to isolate the active devices. The fin height (H_{fin}) depends on the depth of the etching. The source/drain (S/D) junction depth (xj), *i.e.* the depth from the top surface of the fin to the fin body, which is changeable, modulates the current–voltage characteristics. The drain junction is formed by tilted ion implantation. The thick arrow marks in the schematic represent the direction of the stress due to the SiN layer at high temperature. It seems that the tensile stress induced by the SiN expansion is increased with the increase in thickness from the direction of the nitride expansion at high temperature. Further, in the bulk FinFET

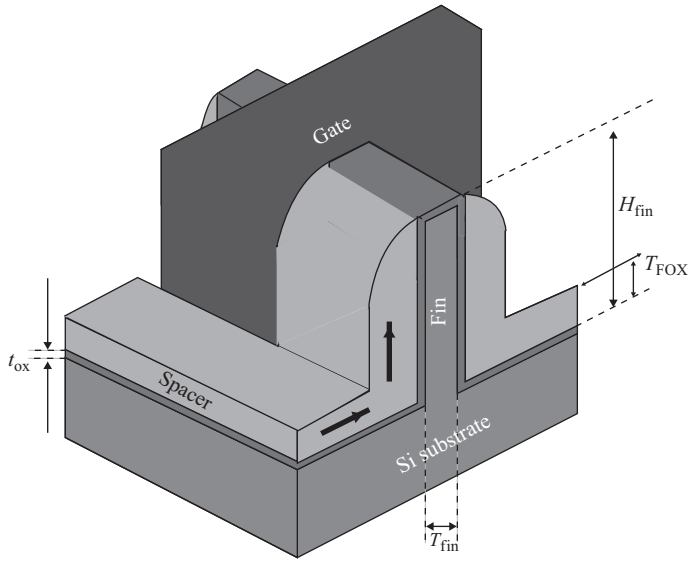


Figure 3.2 Bulk FinFET device structure

device, both the side channel of the body and the body near the junction depth have upward stress [12].

In the bulk FinFET device, the ON-current (I_{ON}) is normalized to the device width (W_{eff}) resulting in an improved FinFET due to lightly doped channel. The source/drain lateral doping gradient (2 nm/dec) and retrograde well doping gradient (4 nm/dec) are same for the tri-gate and FinFET devices [13]. The total gate capacitance is larger for the FinFET device due to larger fringing capacitances. Since the FinFET has a narrower and taller stripe, the thickness and height of the gate electrode along the channel-stripe sidewalls are larger for the FinFET structure that results in larger outer fringing capacitance (C_{of}). Similarly, the gate-to-substrate coupling capacitance is also large for the FinFET device [14]. The bulk-FinFET has the reduced short-channel effects such as subthreshold current and DIBL by reducing electric field effect between the source and drain depicted in Figure 3.3. The drain field effect is also reduced due to lightly doped drain thereby reducing the GIDL effect. The electric field through bulk is still present in bulk FinFET that causes subthreshold current between source and drain as illustrated in Figure 3.3.

The gate-controlled electric field across the channel should be symmetric for the symmetric threshold voltage along the channel. Top and bottom corners of the fin produce additive electric field from top and side walls. Therefore, the corner channel is still ON in OFF device state as depicted in Figure 3.4. This shows charge sharing effect across the top and bottom corners. The small variation in sizing of the fin due to process variation may further increase the corner effects in the bulk FinFET device.

The bulk FinFET is a cost effective device as it allows the possibility of coexistence with the conventional single-gate bulk MOSFET on the same wafer. Over

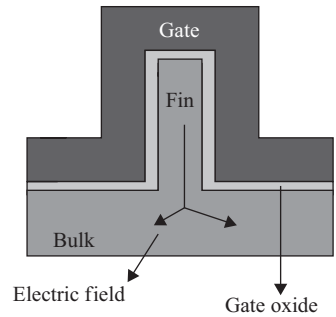


Figure 3.3 Short-channel effects in bulk FinFET device

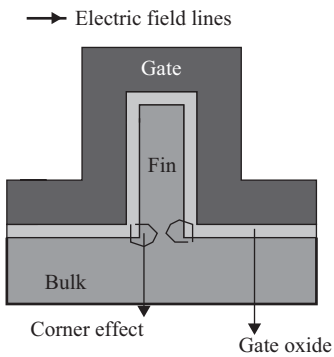


Figure 3.4 Corner effects (charge sharing) in bulk FinFET device

the years, SOI technology has been developed to improve the performance of bulk technology. The main difference between bulk and SOI substrate is the BOX layer below the active silicon layer. The transistor can operate in partially depleted or fully depleted mode based on the silicon thickness used for the SOI wafer. Also, the use of SOI material makes the fabrication of FinFET device easier.

3.2.2 SOI FinFET

In SOI FinFET, an insulating layer is created over the hard mask using lithography and a silicon fin over it using deposition technique. The fully depleted SOI (FDSOI) FinFET device improves electrostatic coupling between the gate and channel. This results in improved linearity, subthreshold slope, body coefficient and current drive capability. The FDSOI technology is useful for various applications ranging from low-power to radio frequency designs [15]. Most notable SOI FinFET devices developed by the researchers are shorted-gate FinFET (SG-FinFET) and independent-gate FinFET (IG-FinFET) [3]. These FinFET devices are described in the following subsections.

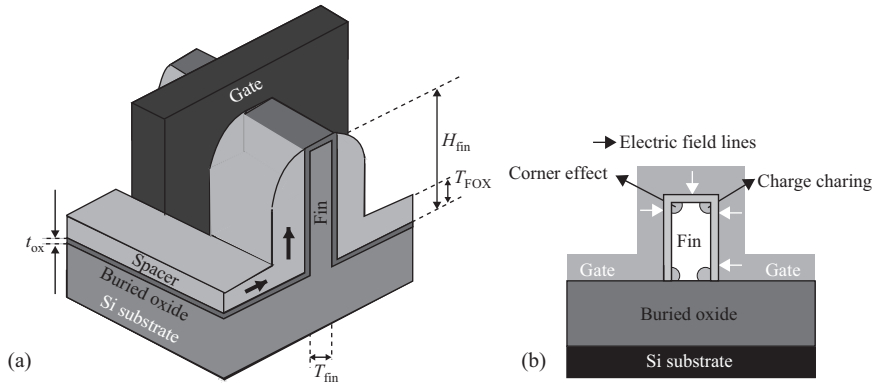


Figure 3.5 (a) SG-FinFET structure and (b) corner effects in SG-FinFET

3.2.2.1 Shorted-gate FinFET (SG-FinFET)

In SG-FinFET, the conducting channel is wrapped up by a thin silicon fin that forms the body of the device. The thickness of the fin is measured in the direction from source to drain that determines the effective channel length of the device. The wrap-around gate structure provides an improved electrical control over the channel resulting in the reduced leakage current and therefore, overcomes other short-channel effects. The device structure of the SG-FinFET is depicted in Figure 3.5(a). This is basically a three-gate device namely front, back and top, all are connected by single electrode. The SG-FinFET device provides higher ON-current (I_{ON}) and OFF-current (I_{OFF} or the subthreshold current) compared to other FinFET devices. The OFF-current (I_{OFF}) increases due to the corner effects in the SG-FinFET device.

The corner effect is a phenomenon at the edges of the active area adjacent to shallow trench isolation where leakage is especially strong due to induced electric field. Under the corner effect, the device is not completely switched OFF due to additive electric field from top and sidewall gates [16]. This causes the charge sharing effects between the top and sidewall gates. The single-gate observes the presence of two threshold voltages at the corners of top and sidewall Si–SiO_2 interface [17]. The difference in the threshold voltages at the corners reduces the switching performance of the device. The charge sharing effect increases with the increase in chip density that increases the temperature and results in increase in the OFF-current across the corners (Figure 3.5(b)). Further, in the SG-FinFET, the fin extension region that is not under the gate is covered by SiN to isolate the electric fields. This region is known as spacer of the device. However, this is an unavoidable region because the steep lateral doping gradient is not adjacent to both the highly doped source/drain region and the lightly doped channel region. The lightly doped channel is preferred because it reduces the corner effects [16,18], random dopant fluctuations (RDFs) and mobility degradation.

Moreover, the corner effect is found severe in shorted-gate devices and to diminish this effect, top-thick gate oxide is used (Figure 3.6). Hence, when the top gate

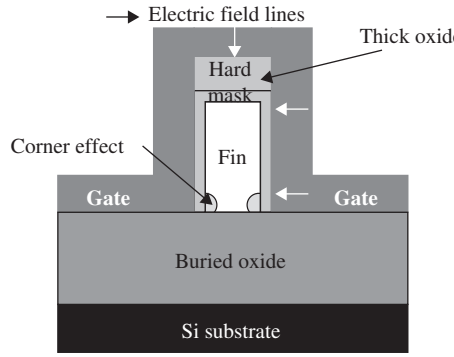


Figure 3.6 Top-thick oxide-based SG-FinFET structure and corner effects

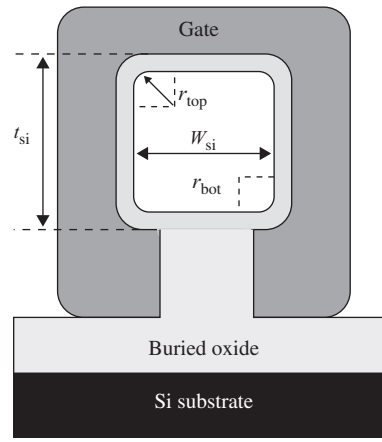


Figure 3.7 Ω-gate FinFET device structure

control is completely eliminated, only front and back gates control the channel. However, the bottom corner effect is still present in the SG-FinFET devices since both the gate oxides are thicker than the fin depth. The problem of corner effects is addressed in the variation of the SG-FinFET device named as Omega-gate FinFET (Ω-gate FinFET) and explained below.

3.2.2.2 Omega-gate FinFET (Ω-gate FinFET)

The Ω-gate FinFET is a single-gate device as shown in Figure 3.7. The thickness and width of the device are t_{Si} and W_{Si} and the radii of the curvature of the top and bottom corners are r_{top} and r_{bot} respectively. In this device also, the conducting channel is wrapped up by a thin silicon fin which forms the body of the device. Thin silicon fin is used as channel for conduction between source and drain, and the single-gate terminal is used to control the device current. The electric field across the gate is symmetric and smooth due to semi-circular structure of the fin. This provides

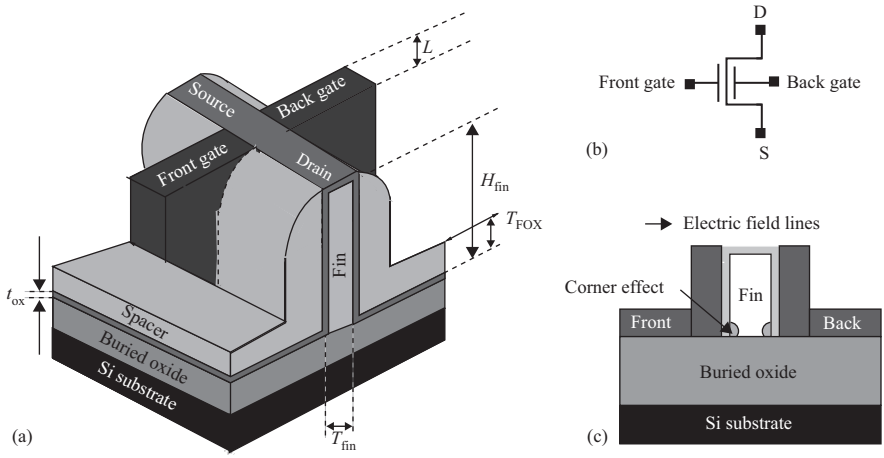


Figure 3.8 IG-FinFET. (a) Device structure, (b) symbol and (c) corner effect

shielding of electric field across the channel, *i.e.* front, back and top sides [19]. Also, the gate is extended under bottom side to overcome the bottom corner effects that results in reduced leakage current. The gate extension of Ω-gate FinFET not only shields electric field lines from the drain for decreasing the corner and DIBL effects but improves the gate-to-channel controllability. With the undercutting of the BOX, the fin resistance decreases that resembles the Pi-gate structure, *i.e.* a triple-gate structure with the gate extending vertically into the BOX [20].

3.2.2.3 Independent-gate FinFET (IG-FinFET)

In IG-FinFET, the top portion of the gate of the SG-FinFET device is etched out, giving rise to two independent gates as shown in Figure 3.8(a). These two independent gates can be controlled separately [21]. This is a four-terminal device: front-gate (G_1), back-gate (G_2), source (S) and drain (D) terminals as shown in Figure 3.8(b) [22,23] for n-channel IG-FinFET. In IG-FinFET device structure, H_{fin} is the height of the fin, T_{fin} is the thickness of the fin, L is the effective device gate length, W is the effective device width, and t_{ox} is the gate-oxide thickness. The effective device gate-length (L) is the distance between source and drain under the poly-silicon. The device width (W) is defined as [24]:

$$W = 2N_{fin}H_{fin} \tag{3.1}$$

where N_{fin} is the number of fins in the IG-FinFET.

In the IG-FinFET device, lightly doped silicon fin is developed over BOX, and it is connected to moderately doped source and drain terminals. The spacers are fabricated using SiN to pattern multiple fins in a single device [25]. The device structure has two poly-silicon gates (front and back) over thin SiO_2 [26]. Since tunnelling of charge carrier increases due to thin gate oxide, high-K dielectric materials are used

in place of SiO_2. These materials reduce tunnelling phenomenon such as band-to-band, direct and Fowler–Nordheim tunnelling between gate and fin [27]. Further, due to non-existence of the top gate, this device reduces the corner effect as shown in Figure 3.8(c). However, few limitations such as difficult fabrication over silicon fin and increased surface scattering reduce the ON-current of the device.

The lightly doped fin in IG-FinFET reduces the short-channel effect and RDF. The device parameters are mostly affected by line edge roughness and line width roughness [24]. The parameters variation changes threshold voltage and ON-/OFF-current of the device that affects the performance and stability of the IG-FinFET-based circuits. The gate sidewall spacer thickness and the work function tuning are used to meet the targeted ON-/OFF-current. The lower work function better controls the channel of the IG-FinFET and reduces the short-channel effects [28]. Moreover, a thinner fin better controls the leakage current and the short-channel effects [29]. The variation in I_{ON} and I_{OFF} currents is comparable in IG-FinFET for both thinner and thicker fins due to the reduced short-channel effects [25,30]. Using BSIM-IMG model, the IG-FinFET device current [9] is given as:

$$I_{ds} = \frac{\mu_{eff}}{\sqrt{1+\theta_{sat}\left(\psi_{s1,D}-\psi_{s1,S}\right)^2}}\frac{W}{L} \times\left\{\eta v_{th}\left(Q_{inv,S}-Q_{inv,D}\right)+\frac{Q_{inv,S}+Q_{inv,D}}{2}\left(Q_{inv,S}-Q_{inv,D}\right)\right\} \tag{3.2}$$

where μ_{eff} is the effective mobility of electrons, Q_{inv} is the inversion carrier density, ψ_{s1} is the front-side surface potential, vth is the thermal voltage, θ_{sat} and η are given by:

$$\theta_{sat} = \frac{\mu_{eff}}{v_{sat}L}\mu_{eff} \tag{3.3}$$

$$\eta = 2-\frac{\epsilon_{si}\overline{E}_{s2}}{Q_{inv}+2\epsilon_{si}\overline{E}_{s2}} \tag{3.4}$$

where ϵ_{Si} is the permittivity of silicon, and $\overline{E}_{s2}$ is the average back-gate electric field. The ON-current increases with the increase in the device width (W), and the effective current decreases with the increase in device gate length (L).

The IG-FinFET device has several advantages over other FinFET devices and found quite effective for low-power and high-performance IC designs including SRAMs. Various SRAM topologies based on IG-FinFET devices have been worked out for designing low-power, high-performance and reliable SRAMs. These topologies are discussed in the next section.

3.3 FinFET-based SRAM topologies

Several SRAM topologies have been developed in the past keeping in view of the conventional as well as recent MOS technologies. This section presents FinFET-based

SRAM topologies due to significant advantages of FinFET devices over the MOFETs including the continued scaling and the device operation at lower V_{DD}. This leads to a considerable improvement in the stability in terms of static noise margin (SNM), read noise margin (RNM) and write margin (WM) at low V_{DD} [31]. Further, FinFET-based SRAMs provide a better signal-to-noise ratio and reliability as compared to the conventional MOSFET-based SRAMs [24,32].

Among the conventional MOSFET-based SRAM topologies, namely, 4T, 6T, 8T and 10 T SRAMs are most popular and widely used in commercial products. FinFET-based SRAM topologies are also realized in the same fashion and classified as 4T, 6T and so on. In FinFET technology, 6T SRAM cell is used as a conventional design. However, IG-FinFETs are used for designing low-power, high-performance and stable SRAMs. The IG-FinFET-based 6T SRAM cell topology is discussed in the following subsection.

3.3.1 IG-FinFET-based 6T SRAM

The IG-FinFET-based 6T SRAM cell topology consists of two cross-coupled inverters and is shown in Figure 3.9 [32]. Two pull-up transistors (PU_1 and PU_2) and two pull-down transistors (PD_1 and PD_2) are employed to design the cell inverters. The cross-coupled inverters create a loop that is used to latch the bit in the corresponding state. In addition, two access transistors (A_1 and A_2) are used to access the bit from the SRAM cell. The wordline (WL) and bitlines (BL and BLB) are needed to access the bit data from the cell. In latch '1' state, the transistors PU_1 and PD_2 will be turned ON, and the transistors PU_2 and PD_1 will be OFF. This helps to store the logic values ($Q =$ '1' and QB = '0'). The wordline (WL) is used to drive the access transistors (A_1 and A_2), and keeping the wordline low (*i.e.* WL = 0) disconnects the SRAM cell

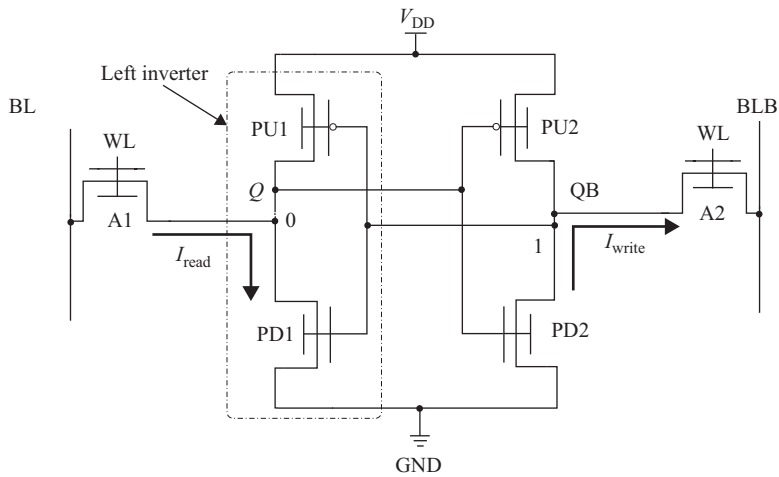


Figure 3.9 IG-FinFET-based 6T SRAM cell

from the bitlines (BL and BLB). These bitlines remain precharged in standby mode. The cell read and write operations are discussed below.

3.3.1.1 Read and write operation

In read operation, it is assumed that prevailing state of SRAM is '0'. The bitlines are first precharged to V_{DD} and then enable the wordline to logic '1' that turns the access transistors A_1 and A_2 ON. The voltage of column BLB retains its precharge level, while A_1 and PD_1 transistors pull down the voltage of BL. The data-read circuitry detects small voltage difference between BL and BLB that decides logic '0' as the data output. The current path from the access transistor (A_1) to the pull-down transistor (PD_1) is called read current path as shown in Figure 3.9. The read performance improves with the increased read current (I_{read}), whereas the propagation delay of this operation is called read access time (T_{read}).

In write operation, the input drivers, which write the data into the memory, activate the bitlines. To write '1' into the cell, BL is charged up to V_{DD} and BLB discharged up to GND. By enabling the worldline, the access transistors (A_1 and A_2) turn ON. The pull-down transistor (PD_1) turns OFF and the pull-up transistor (PU_1) turns ON. The voltage at node Q attains logic high, while QB goes low. The path from the pull-down transistor (PD_2) to the access transistor (A_2) forms the write current path to write '1'. The write performance is improved with the increase in write current (I_{write}), and the propagation delay in writing the bit into the SRAM cell is called write time (T_{write}).

3.3.1.2 SRAM cell design

Transistor sizing is critical to the SRAM cell design as it plays key role in its correct operation and also improves the stability of the cell in standby and read operations. In read '0', the node voltage Q will increase from its initial value of '0'. The node voltage Q can rise to the threshold voltage of PD_2 transistor during the read operation, forcing an unintended change in the stored state. However, the voltage must not exceed the threshold voltage of PD_2, so that the transistor PD_2 remains turned OFF during the read operation, *i.e.*

$$V_{Q_{max}} \leq V_{th_{PD2}} \tag{3.5}$$

where $V_{Q_{max}}$ is the maximum voltage at node Q, and $V_{th_{PD2}}$ is the threshold voltage of the PD_2 transistor. The amount of the current (I_{ds}) flowing through the transistor depends on its size, *i.e.* (W/L) ratio and is given by following IG-FinFET device current [8,9] equation:

$$I_{ds} = \mu_0 \frac{W}{L}\left[\frac{Q_{inv,s} - Q_{inv,d}}{2}\left(\psi_{s1.d} - \psi_{s1.s}\right) + \eta\frac{KT}{q}\left(Q_{inv,s} - Q_{inv,d}\right)\right] \tag{3.6}$$

where μ_0 is the mobility of electrons, T is the temperature, Q_{inv} is the inversion carrier density, ψ_{s1} is the front-side surface potential, q is the charge of electron, and η is given by:

$$\eta = 2 - \frac{\epsilon_{\text{Si}}\overline{E}_{s2}}{Q_{\text{inv}} + 2\epsilon_{\text{Si}}\overline{E}_{s2}} \tag{3.7}$$

where ϵ_{Si} is the permittivity of silicon, and $\overline{E}_{s2}$ is the back-gate side electric field. The read stability of the SRAM cell increases with the increase of β ratio, *i.e.* by increasing the strength of pull-down transistors where β ratio is defined by

$$\beta = \frac{\left(\frac{W}{L}\right)_{\text{PD}}}{\left(\frac{W}{L}\right)_{A}} \tag{3.8}$$

The transistor size is also critical to improve the write performance. Let us consider the previous state of SRAM is logic '1' and to change the stored information, *i.e.* $Q = 0$ and $\text{QB} = 1$, the node voltage Q must be reduced below the threshold voltage of PD_2 transistor to turn it OFF. When the transistors A_1 and PU_1 are turned ON, the data will be stored in the SRAM cell. The α ratio affects the write operation of the SRAM cell and is defined by

$$\alpha = \frac{\left(\frac{W}{L}\right)_{A}}{\left(\frac{W}{L}\right)_{\text{PU}}} \tag{3.9}$$

In IG-FinFET, the transistor width (W) is a function of number of fins (N_{fin}) and fin height (H_{fin}), *i.e.* $W = 2\ H_{\text{fin}} \times N_{\text{fin}}$. The device gate length and fin height are fixed according to the process technology, and N_{fin} can be calculated by the designer and used for the desired read and write operations. Carlson *et al.* [33] proposed back-gate bias IG-FinFET-based SRAM cell topology. This topology is described in the following section.

3.3.2 Back-gate bias IG-FinFET-based 6T SRAM

The back-gate bias IG-FinFET-based 6T SRAM cell that enhances read and write performances is shown in Figure 3.10. The back-gate biasing is employed to optimize the performance of IG-FinFET-based SRAMs through a dynamic adjustment of the effective α and β ratios. In this topology, the back-gates of the access transistors (A_1 and A_2) are connected to the storage nodes (Q and QB) of 6T SRAM cell. This mechanism tunes the gate-work function of the access transistors [33]. If the stored bit in SRAM is '0', then Q and QB will be at logic high and low levels, respectively. The back-gate of the access transistor A_1 connected to Q is bias at '0' voltage, decreasing the strength of the access transistor and increases the β ratio of the SRAM cell for read operation. This forces the access transistor to keep the storage node at '0'. The β ratio is maintained throughout the access and standby mode thereby increases the RNM.

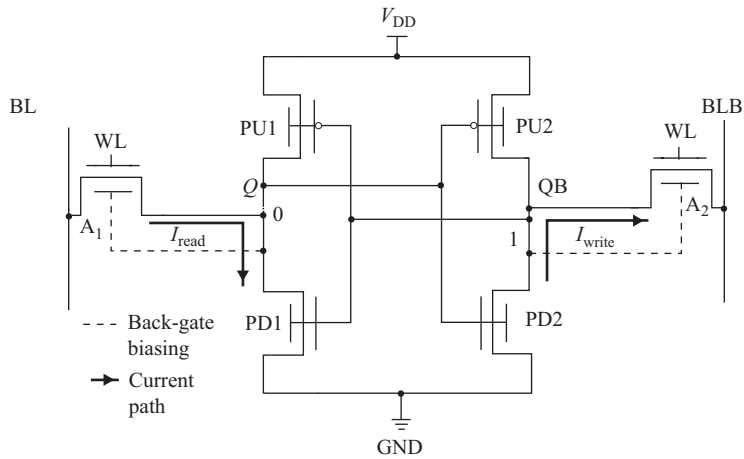


Figure 3.10 Back-gate bias IG-FinFET-based 6T SRAM cell (Source: Carlson, 2006, p. 2 [33])

Consider storage nodes *Q* and QB are at logic '0' and '1', respectively, prior to write operation. In this case, strength of the back-gate bias transistor (A_2) increases which helps in discharging the storage node QB for flipping the bit. However, the writability increases with strong access and weak pull-up transistors. Moreover, the back-gate bias 6T SRAM cell provides improved RNM by tuning the gate-work function of the access transistors and also exhibits reduced sensitivities to process variations that results in the yield improvement. The increased process variations and low-power requirements of SRAMs led the designers to develop new topologies, and one such topology is PPN 10T SRAM. The IG-FinFET PPN 10T SRAM topology is described in the following subsection.

3.3.3 IG-FinFET-based PPN 10T SRAM

Basically, PPN-based 10T SRAM topology has feedback loop to restore the storing nodes during read and standby mode of operations (MOPs) [22,34]. This topology provides good tolerance towards process variation-aware SRAM designs with increased stability. The IG-FinFET-based PPN10T SRAM cell topology is shown in Figure 3.11.

This PPN 10T SRAM cell consists of two cross-coupled P-P-N inverters. The data stored at nodes *Q* and QB are complement to each other. The transistors PU_1, PI_1 and PD_1 form a left-side P-P-N inverter, and the transistors PU_2, PI_2 and PD_2 form the right-side inverter. The transistors A_1 and PI_1 or the transistors A_2 and PI_2 form the restoring path depending upon the logic value stored at *Q* and QB. The restoring path formed by the transistors A_2 and PI_2 as shown in Figure 3.11 is the case of QB at logic value '1'. The transistors A_1 and R_1 or A_2 and R_2 provide the read current (I_{read}) path for discharging the bitlines (BL or BLB) depending on the stored values at *Q* or QB. The source terminals of the read transistors R_1 and R_2 are

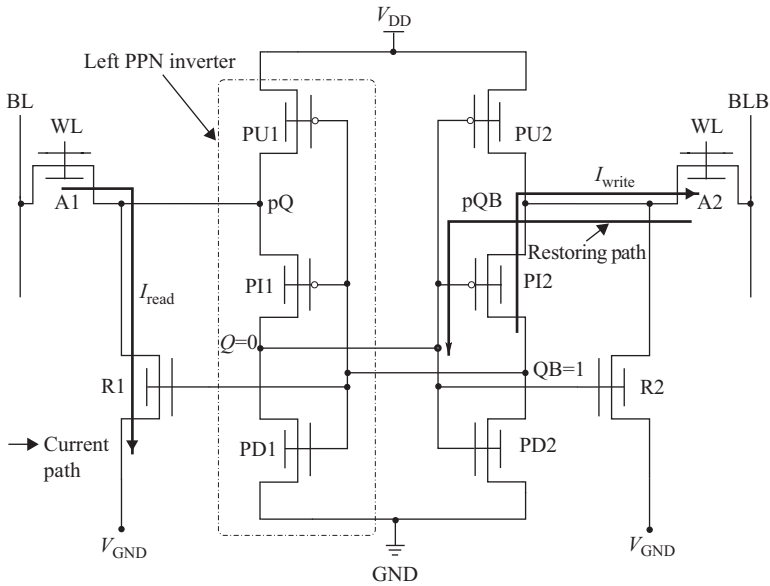


Figure 3.11 IG-FinFET-based PPN 10T SRAM cell (Source: Yadav, 2015, p. 4 [36])

connected to the virtual ground (V_{GND}). In read operation, the V_{GND} is connected to the ground (GND), and for other operations (write and standby), the V_{GND} is connected to V_{DD} to reduce the leakage current. Further, the pull-up and intermediate transistors are connected to the nodes (pQ and pQB), and they are known as the pseudo storage nodes. Each of the pseudo storage nodes is located between the two cascaded pFinFET transistors forming the P-P-N inverter. During the read operation, these nodes provide an isolation mechanism between the bitline pair and the true storage nodes (Q and QB) that results in reduced data-dependent bitline leakage.

In read operation, the bitlines (BL and BLB) are first precharged to V_{DD} and then enable the wordline (WL) while V_{GND} is connected to GND. The access transistors (A_1 and A_2) are turned ON, and one of the read transistors (R_1 or R_2) will be ON according to the stored bit in the SRAM cell. The small voltage difference between bitlines, BL and BLB, allows to read the store bit, and the cell provides isolated read from the storage nodes. The intermediate transistors (PI_1 and PI_2) form the feedback path to restore the nodes in order to improve the read stability of the SRAM cell. In write '1' operation, the bitline BL is first precharged to V_{DD}, and the bitline BLB is discharged, and then enable the worldline (WL) to turn ON the access transistors. The pull-up transistor PU_1 and the pull-down transistor PD_1 will be turned ON and OFF, respectively, to store the bit. However, the intermediate transistors (PI_1 and PI_2) limit the write performance by providing prevailing isolation path between pseudo and storage nodes.

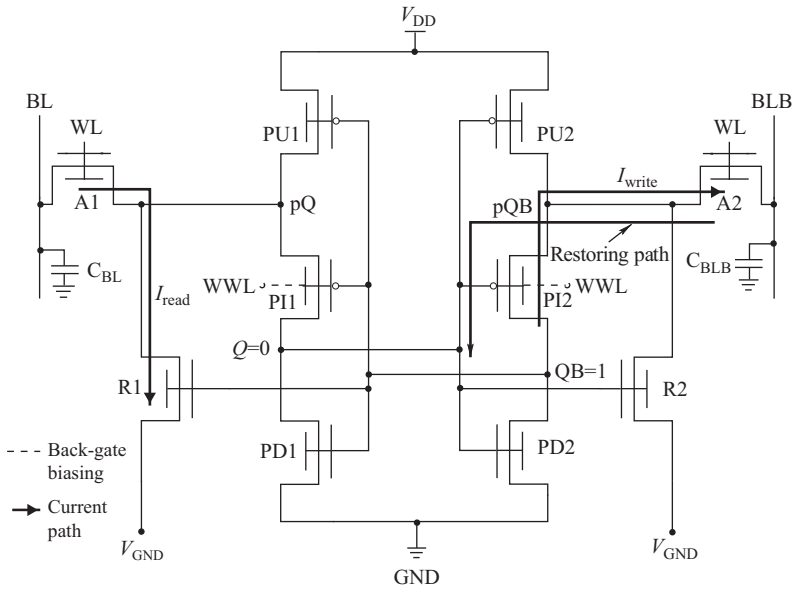


Figure 3.12 IG-FinFET-based back-gate bias PPN 10 T SRAM cell (Source: Yadav, 2013, p. 2 [22])

The write performance and increased process variations issues have been addressed by Yadav *et al*. [22,35] and proposed different IG-FinFET-based PPN 10 T SRAM topologies, namely back-gate bias 10 T, cross-coupled back-gate bias 10 T and hybrid back-gate bias 10 T topologies. All these IG-FinFET-based SRAM topologies are discussed in the subsections to follow.

3.3.3.1 Back-gate bias PPN 10 T SRAM

The IG-FinFET-based back-gate bias PPN 10 T SRAM cell topology is shown in Figure 3.12. This cell utilizes differential sensing with 10 T transistors, two wordlines (WL and WWL) and two bitlines (BL and BLB). Two intermediate transistors (PI_1 and PI_2) are employed to restore the storage node values during cell operation and raise the switching threshold of the inverter during the read operation. The read transistors (R_1 and R_2) are active by V_{GND} signal only in read operation. This strengthens the proposed SRAM topology for subthreshold read operation and reduces the power budget. The WWL signal enables in write mode, and it provides strength to the intermediate transistors (PI_1 and PI_2) thereby increases the write performance. Both the intermediate transistors are moderately ON during write mode that results in direct conducting path between access and storage node. This considerably improves the write performance as well as the stability of the SRAM cell.

The stability of SRAM is a function of α and β ratios as WM and RNM are proportional to α and β, respectively. Since the device current is a function of W, α

and β can be represented in terms of the device current for the back-gate bias SRAM topology as

$$\beta = I_R/I_A \tag{3.10}$$

$$\alpha = I_A/((I_{PU} + I_{PI}) \tag{3.11}$$

where I_A, I_R, I_{PU} and I_{PI} are the current flowing in access, read, pull-up and intermediate transistors, respectively. The maximum read current is flowing through access and read transistors (neglecting I_{PI}) that results in increased β ratio. The intermediate transistors are moderately ON. Hence, the stability of read and write is increased collectively in case of the increased α ratio as compared to the conventional 6T and PPN 10 T topology. However, this topology needs one additional control line, *i.e.* WWL, resulting in increased layout of the SRAM. To overcome this problem, new SRAM cell topology namely cross-coupled back-gate bias 10 T is proposed by Yadav *et al.* [22] and is described in the next subsection.

3.3.3.2 Cross-coupled back-gate bias PPN 10 T SRAM

This IG-FinFET-based 10 T SRAM cell topology has different (cross-coupled) feedback mechanisms over back-gate bias 10 T SRAM cell as shown in Figure 3.13 [35]. The intermediate transistors are cross-coupled that provides feedback in order to improve read/write performance of the SRAM cell. These transistors (PI_1 and PI_2) are ON based on the data input that results in storing the bit in small time interval. This shows unique write operation dependent on the data input. However, in read mode, the back-gates of the intermediate transistors are completely OFF thereby, increases strength of the inverter to improve the RNM of the SRAM cell. Other operations namely standby and write are similar to the previous SRAM cell [22,35]. This SRAM cell topology is further modified to enhance read/write performance and improvement in its stability. The modified IG-FinFET-based 10 T SRAM topology is described next.

3.3.3.3 Hybrid back-gate bias PPN 10 T SRAM

In hybrid back-gate bias PPN 10 T SRAM cell topology, the back-gate bias approach is combined with the cross-coupled back-gate bias approach. Figure 3.14 [36] shows the schematic of hybrid back-gate bias 10 T SRAM cell topology. The additional back-gate biasing in the pull-up and pull-down transistors improves write performance along with stability. The back-gate of these four transistors will be OFF in the entire operation cycle. Hence, this also saves significant power of the SRAM cell. The read and write operation of this SRAM topology is similar to the back-gate bias 10 T SRAM topology. In hybrid topology, both gates of the access transistors (A_1 and A_2) are enabled (*i.e.* connected to V_{DD}) in read operation that provides sufficient RNM. In this topology, the cell operations such as standby and write are also similar to the previous 10 T SRAM topologies.

The SRAM cell stability is one of the major issues addressed in all the cell topologies discussed so far. The key factors that decide the cell stability are SNM, RNM

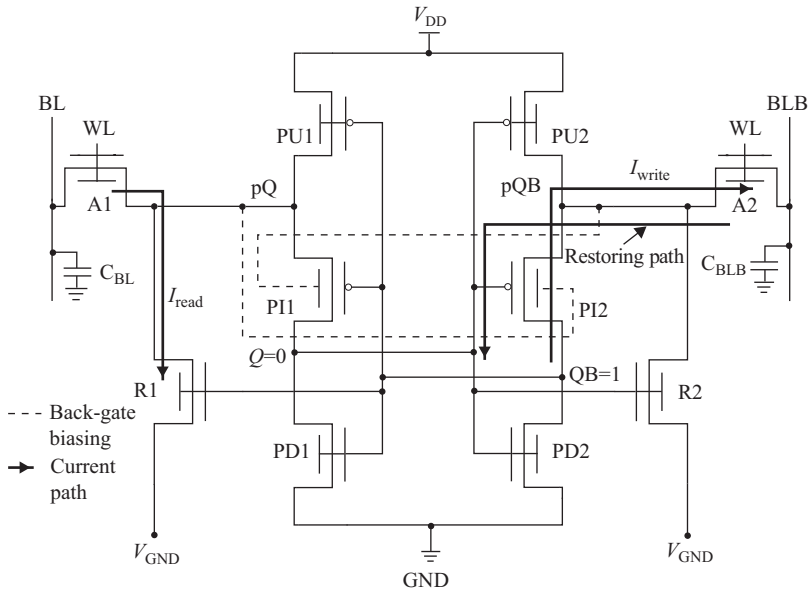


Figure 3.13 Cross-coupled back-gate bias PPN 10T SRAM cell (Source: Yadav, 2013, p. 2 [35])

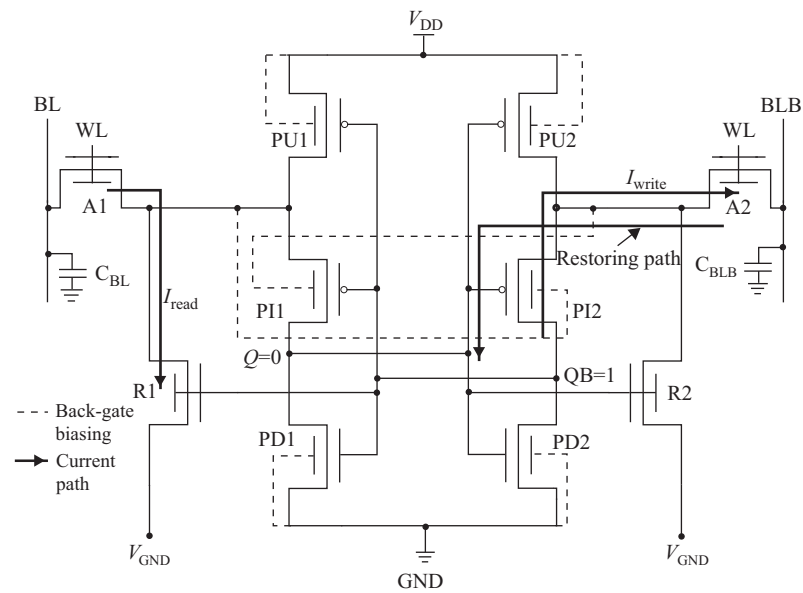


Figure 3.14 Hybrid back-gate bias PPN 10T SRAM cell topology (Source: Yadav, 2015, p. 5 [36])

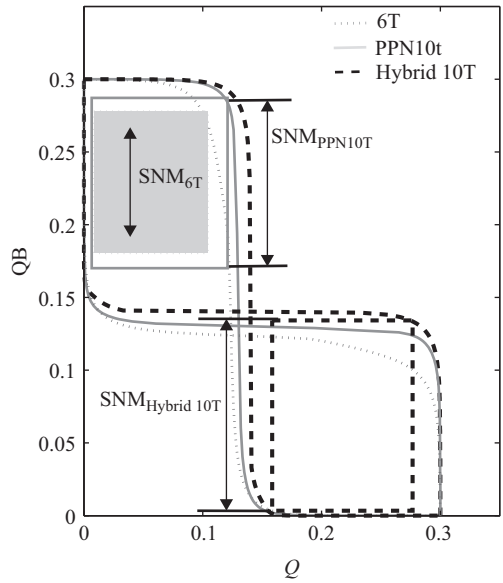


Figure 3.15 Butterfly curve of the cross-coupled inverter for SRAM cell topologies

and WM. These factors are governed by the device parameters of all the transistors employed in a given SRAM cell topology. Further, data retention in standby and read modes is an essential requirement of the SRAM, and the cell stability plays an important role for the correct operations of the SRAM cell. The stability analysis of the above IG-FinFET-based SRAM topologies is given in the following subsection.

3.3.4 Stability analysis

The cell stability is defined as the maximum DC noise voltage (V_n) that its topology can tolerate without changing the present stored bit. In the SRAM cell, it is measured in terms of SNM, RNM and WM. The stability of the various IG-FinFET-based 6 T and 10 T SRAM cell topologies is analysed through simulations using 32 nm Predictive Technology Model (PTM) [37] at 300 mV supply voltages [36]. The butterfly curve of the cross-coupled inverter for different SRAM topologies is shown in Figure 3.15. The curve clearly shows that the hybrid PPN10 T cell provides improved SNM over other 6 T and 10 T cells. Figure 3.16 [22,35,36] gives the comparative analysis in terms of WM, SNM and RNM for 10 T [38], PPN 10 T and hybrid back-gate bias 10 T SRAM cells.

The simulation results show that the effect of process is increasing rapidly with the technology advancement, and it is not limited to ∼10% [30] as was the case in the conventional CMOS technology. The process variations ultimately degrade the yield and reliability of the SRAM due to decreased RNM, SNM, etc. In order to analyse the cell stability under process variation, MC simulation for 5000 points with $6\sigma = 10\%$

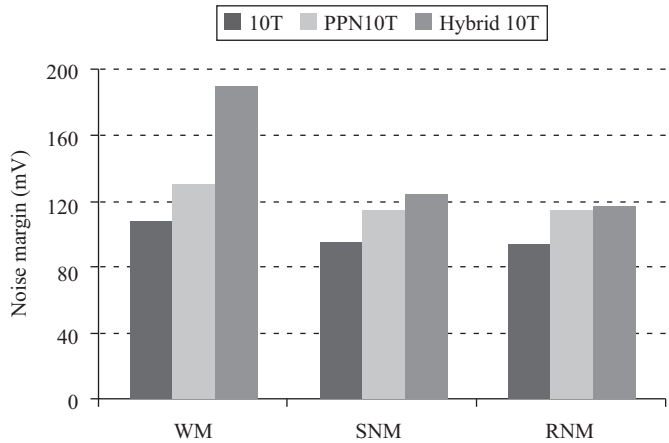


Figure 3.16 WM, SNM and RNM for IG-FinFET-based 10 T SRAM cell topologies (Source: Yadav, 2015, p. 5 [36])

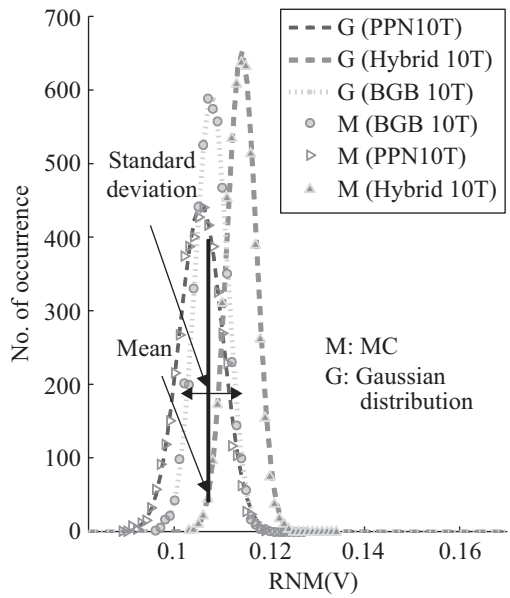


Figure 3.17 RNM distribution for various IG-FinFET-based SRAM cells (Source: Yadav, 2015, p. 11 [36])

deviation (variation) for statistical analysis is conducted. The Gaussian distribution plots for RNM with process variations (intra-die) are shown in Figure 3.17 [36]. These plots demonstrate the variation in the RNM of the PPN10 T, back-gate bias 10 T, cross-coupled back-gate bias 10 T and hybrid back-gate bias 10 T SRAM cells. It can be further observed that the standard deviation of the hybrid back-gate bias

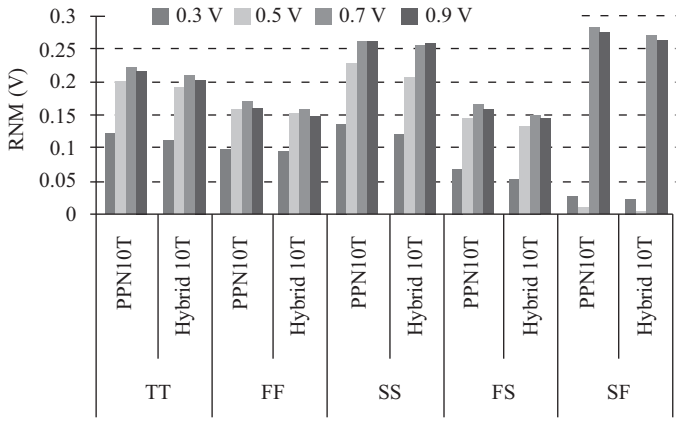


Figure 3.18 RNM at different process corners (PCs) for IG-FinFET-based SRAM cells (Source: Yadav, 2015, p. 12 [36])

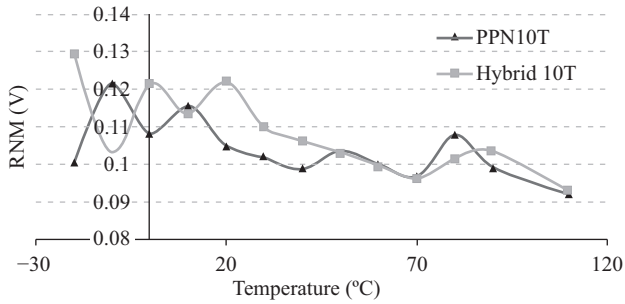


Figure 3.19 RNM with varying temperature for IG-FinFET-based PPN10T and hybrid 10T SRAM cells (Source: Yadav, 2015, p. 12 [36])

SRAM cell is smallest among all the topologies. This exhibits higher tolerance in case of the hybrid back-gate bias SRAM cell towards process variations as compared to other topologies.

Under intra-die process variations, the IG-FinFET-based SRAM topologies have been evaluated and plotted the RNM at each Process Corner (PC) with varying supply voltages for different topologies are evaluated as shown in Figure 3.18 [36]. These results show that the hybrid back-gate bias SRAM cell provides improved (higher) RNM over other 6T and PPN10T SRAM cells. The improved RNM for the hybrid back-gate bias cell topology is due to the presence of the restoring path. Further, since RNM also depends on temperature, the simulations have been carried out to observe the behaviour of RNM under the influence of temperature. The simulation results obtained are shown in Figure 3.19 [36] that clearly demonstrated that the temperature variation effect on the hybrid SRAM cell topology is less as compared to other PPN10T topologies.

The PVT analysis for 6 T, 10 T (proposed by Ebrahimi [38]) and PPN 10 T SRAM cell show that the hybrid back-gate bias 10 T cell exhibits improved tolerant towards PVT variations. Hybrid back-gate bias 10 T SRAM cell mitigates process variation effectively, whereas it shows marginal improvement with respect to voltage and temperature variation due to the high hot spot, self-heating and hot carrier injection (HCI) effect in FinFET. The FinFET-based SRAM cell design challenges are discussed in next section.

3.4 FinFET-based SRAM design challenges

FinFET-based SRAMs deliver various benefits over the conventional MOSFET-based SRAMs due to the un-doped fully depleted channel (fin). However, there are many design challenges due to the novel FinFET device, its critical layout and the effect of PVT variations. The important design challenges of such SRAM cells are the optimization of device dimension (optimized α and β ratios) for successful read and write operation, stable and reliable design for scaled supply voltage, implementation of symmetric device, alignment of vertical square shape fin, physical layout issue to match FinFET device width and realization of long-channel device for read/write buffer [39,40]. Due to PVT variations, the design issues are critical in the design of SRAM cell because of the nanoscale device dimension and high memory density.

The reliable and stable SRAM requires fine geometry of nanoscale FinFET devices. The FinFET device dimensions such as fin thickness (T_{fin}), fin height (H_{fin}), fin length (L_{fin}) and gate-oxide thickness (T_{OX}) are process sensitive, and they are affected by LER, HCI and RDF [24,41]. This results in mismatch of the device threshold voltage, ON-current and OFF-current thereby degrading the performance and reliability of the FinFET-based SRAM. Therefore, process variation tolerant FinFET-based SRAM design is a major issue to decide the device sizing in order to improve performance and stability of the SRAM.

Read, write and standby operation of SRAM depends on device sizing and symmetric pull-up and pull-down network devices. The cell ratios α and β are defined by device sizing for correct operation. Strong access transistor increases the write performance, but decreases β. Weak pull-up transistor also increases the write performance, but decreases the read stability [42]. Therefore, a stable FinFET-based SRAM design requires appropriate α and β ratios especially for low-power SRAM design. However, low-power designs require scaled supply voltage to reduce power dissipation, and the supply voltage scales with continuous scaling of device dimension to maintain the device electric field. The low supply voltage is highly sensitive to the logic level faults, environmental variations and electromagnetic effects thereby reducing the stability and performance of FinFET-based SRAM. Therefore, SRAM cell design for lower supply voltage is an important design challenge where the varying supply voltage is used to improve performance of the SRAM.

Mostly, the supply voltage scaling and wordline under/over-drive voltage are used to increase the read and write performance. In case the wordline is driven over V_{DD} the write performance increases, whereas RNM decreases. Moreover, the under-drive

wordline improves RNM but reduces the read current and read performance. Thus, a design trade-off is made between read performance and stability in FinFET-based SRAM design [43,44]. The important design issues are simultaneous improvement in performance and stability of FinFET-based SRAM and the read-write conflict that increases due to V_{th} mismatch and device scaling.

The continuous device scaling decreases channel area but increases V_{th} resulting in V_{th} mismatch. Since RNM and WM are sensitive to V_{th} variation because of the cell ratios α and β, they greatly affect the read and write performance. The higher β ratio improves the RNM that helps in improving the read performance, whereas α ratio for write operation is improved with decreased conductance of pFinFETs. Moreover, threshold voltage of pFinFET increases due to self-heating in high-density SRAMs. This results in reduced read performance and improved write performance. Therefore, temperature variation-aware SRAM design is a challenge that requires to resolve read-write conflict issue [39].

The FinFET device geometry itself provides various challenges for SRAM circuit and layout design due to vertically placed square-shape fin for conduction. The square-shape fin increases fringing capacitances and causes corner effects due to charge sharing at the corners. In the FinFET-based SRAM cell layout, the cell width should be matched with the fringe fin pitch to provide a smooth shift from the SRAM array to the peripheral circuit [39,45]. Therefore, its layout is critical in FinFET-based SRAM design. Further, the realization of long-channel device for peripherals (read/write buffer) is difficult due to lithographic limitations. In short channel, transistors stacking in series is a major challenge in FinFET-based designs for high performance.

Reliability is a major concern in FinFET-based SRAM designs due to un-doped fins in FinFET device that increases the HCI between fin and gate. High-K dielectric material as gate oxide is used that controls tunnelling current between gate and body but it limits the ON-current (I_{ON}). As a result, self-heating of the FinFET devices increases under dense SRAM architecture. Negative bias temperature instability and positive bias temperature instability also affect the performance of FinFET-based SRAM circuits [46,47], as they shift the device threshold over a period of time and degrade the reliability.

Many of the above design challenges have been addressed in various novel FinFET-based SRAM cell topologies proposed by the researchers. However, PVT variations in the scaled FinFET devices have not been fully concentrated in the design of the SRAM topologies. Moreover, PVT variations in FinFET-based SRAM cell design require additional circuits to be incorporated in order to sense the variations. However, in variation-aware SRAM design, the objective of the designer is not only to detect the variation but also to provide an efficient mitigation approach. The PVT-aware SRAM design and various mitigation techniques are described in the next section.

3.5 PVT-aware SRAM design

Process variation has become a major concern in the design of high-performance nanoscale SRAM. The process parameter variation results in variation in speed and

power consumption in fabricated dies. Moreover, the temperature along with supply voltage variation increases in the high-density SRAM architecture and the sensitivity of circuit parameters increases with reduction in supply voltage. Nanoscale memory cell is more sensitive to device variations causing device mismatch for various reasons. The standard approach to mitigate PVT variations is to develop novel FinFET SRAM designs [48] along with mitigation techniques. Some of the mitigation techniques for reliable FinFET-based SRAM designs are discussed below.

3.5.1 PVT mitigation techniques

Mitigation techniques for PVT-aware FinFET-based SRAM designs first detect the parametric shift due to random variations and then mitigate the variation effects. Due to process variation, shift in the FinFET device parameters degrades the stability and performance of the SRAM. Further, temperature variation causes variation in leakage current, propagation delay and stability. The supply voltage variation also results in significant changes in the device parameters. These sources of parameter variations (PVT) are observed by detecting the variation in the device threshold voltage and leakage current. Sensors are required to measure the leakage current, propagation delay and threshold voltage. The runtime monitoring of these parameters detects the variation in performance and stability of the SRAM due to PVT variations. After detection of the parameters variation, mitigation can be applied using supply voltage scaling and back-gate biasing. In general, two types of mitigation techniques are used in FinFET-based SRAM design: static (fixed) and dynamic (runtime). Static techniques mitigate the variations during the circuit design according to the expected variations in the design whereas dynamic techniques detect the variations during runtime and apply the mitigation technique. Various static and dynamic PVT mitigation techniques for IG-FinFET-based SRAM are discussed as follows.

3.5.1.1 Static mitigation techniques

In recent years, several static mitigation techniques have been proposed to maximize the immunity against PVT variations [49,50]. Threshold voltage and work function tuning of IG-FinFET are mostly used to develop these techniques [51]. Ebrahimi *et al*. proposed IG-FinFET-based static mitigation approach [52] that utilizes back-gate voltage and maximizes the yield of the SRAM cell. In this approach, the IG-FinFET parametric failure mechanism is first identified via the read, write and access failure mechanism. A model is then developed to predict process variation effects in 45 nm and 55 nm technology-based SRAMs. In this model, SRAM metrics considers read, write and standby, Mode of Operations (MOPs) to optimize the yield in the presence of the process variations. The back-gate voltage is used as a design constraint that affects the subthreshold current. Other design parameters such as threshold voltage, ON-current, read current, read/write stability of SRAM are used to model the optimization function. Particle swarm optimization (PSO) algorithm has been employed to obtain the optimum back-gate voltage.

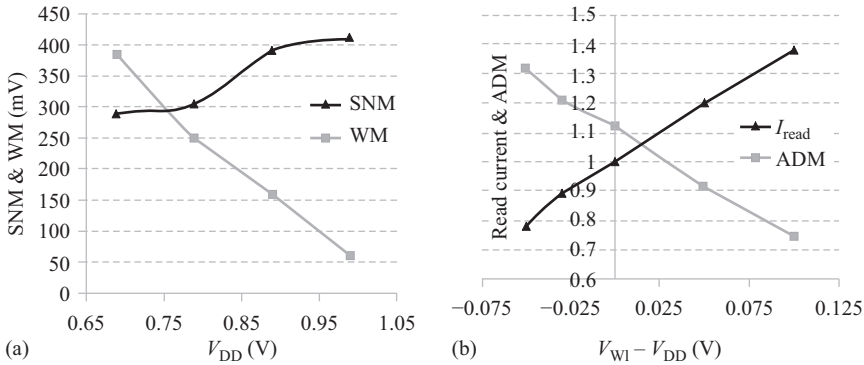


Figure 3.20 (a) SNM and WM versus power supply voltage of IG-FinFET SRAM and (b) read current and ADM versus ($V_{WL} - V_{DD}$) of IG-FinFET SRAM

Supply voltage, bitline and wordline voltages are also used to improve SRAM performance and stability. The variation in SNM and WM for IG-FinFET-based 6T SRAM with respect to varying supply voltage is shown in Figure 3.20(a) [43]. The standby stability is high at higher supply voltage. The storage node voltage of SRAM cell increases at higher supply voltage that decreases the random noise effect. The required low value of WM decreases with reduced supply voltage that improves the write-performance. The storage node voltage is reduced at lower supply voltage that helps in flipping the SRAM cell. Hence, the reduced supply voltage helps in improving the write performance whereas increased supply voltage improves the read stability. The read current (I_{read}) and access disturb margin (ADM) for varying wordline voltages are shown in Figure 3.20(b). The read current increases with increased WL voltage whereas ADM decreases with increased WL voltage. The ADM is improved at the cost of reduced read current for under-drive WL and also by reducing the charge injection from the precharged BL to the low logic ('0') node in the active SRAM cell. This technique can be utilized in designing SRAM with required WM. However, negative bitline (NBL) assist circuit may further enlarge the WM [43,44]. Thomas *et al*. [53] used the NBL approach to improve the SRAM cell stability. In this case, two different wordlines are used that are connected to the front and back gates of each access transistors. An IG-FinFET-based SRAM has also been successfully designed with a considerable leakage reduction [54]. All these static techniques suffer with the prior assumption of worst-case PVT variations and further deviate from the experimental results in real time scenario. Therefore, dynamic mitigation techniques have been proposed for random PVT variations that detect variations in runtime.

3.5.1.2 Dynamic mitigation techniques

Dynamic mitigation scheme for PVT-aware SRAM design shown in Figure 3.21 first detects the variation in SRAM parameters due to PVT variations and then mitigates the variation in FinFET-based SRAM. Selection of SRAM parameters is

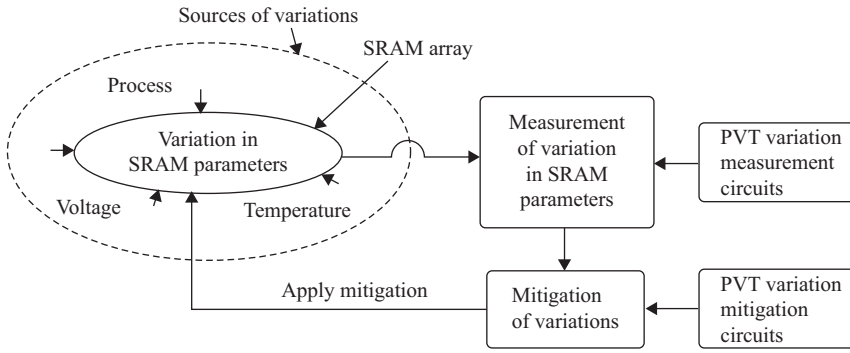


Figure 3.21 Dynamic mitigation for PVT-aware SRAM design

important to consider the variation in the design and to mitigate symmetric variation effect due to individual PVT variations. Leakage current, propagation delay and SNM are most common parameters used in dynamic mitigation techniques at circuit level. Few additional circuits such as current mirror, inverter chain and voltage comparator are also employed to monitor variation in leakage, delay and SNM, respectively.

Most widely used techniques to mitigate the variations in PVT-aware SRAM are back-gate biasing, NBL and dynamic voltage scaling (DVS). The PVT variations in SRAM can be mitigated via supply voltage scaling without adding extra circuit and found very useful in write and standby modes of operation. This technique also provides large amount of power saving at the cost of degraded performance due to extra circuitry. At architecture level, redundant row/column-based techniques have been explored in order to improve yield of the SRAM [55,56]. These techniques detect and replace faulty cells by adaptively remapping SRAM array. Various dynamic PVT mitigation techniques using delay, leakage current, NBL and under/over-drive worldline [43,57–59] are proposed for reliable SRAM design. Most popular and widely used mitigation techniques such as NBL, leakage and inverter delay monitoring have been implemented in various PVT-aware SRAM designs. These designs are presented in the next subsection.

3.5.2 PVT-aware SRAM designs

Various PVT-aware SRAM designs using dynamic mitigation techniques namely, NBL-driven [43,60], leakage-driven [57] and sensitivity-driven techniques [63] are discussed here.

3.5.2.1 NBL-driven design

NBL-driven PVT-aware SRAM block schematic is shown in Figure 3.22. In this design, NBL trigger and negative voltage generator circuit is the main block that plays key role in improving the write operation as shown in Figure 3.23. The logic level of

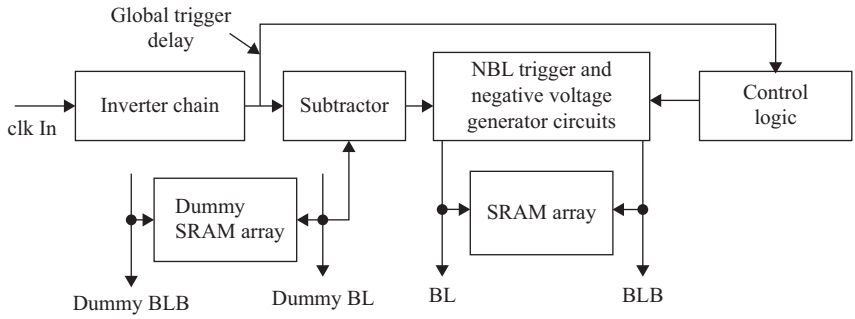


Figure 3.22 Block schematic of NBL-driven PVT-aware SRAM

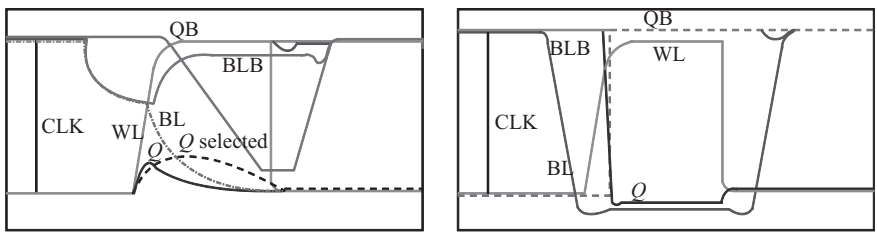


Figure 3.23 Write operation simulation results (a) without NBL and (b) with NBL technique

bitline is pulled-down below the GND using boost capacitor (C_{boost}) just after disabling the precharge in the access MOP. Different voltage levels for NBL can be achieved by programming the trigger delay or the number of sources for injecting the charge in the boost capacitor. Smaller boost capacitor generates a smaller value of charge that results in a smaller voltage level (V_{NBL}) for NBL. The different negative bitline triggering voltage levels are generated by small value capacitors that provide negative levels at different time instances using triggering circuit as shown in Figure 3.24 [60], and the large step size is generated by large delay chain (chain of inverters).

The variation is detected by the global triggering delay circuit, and the delay is provided by the dummy bitline from the dummy SRAM. The difference between delays is used to detect the variations in SRAM due to PVT variation. Furthermore, a fixed inverter chain is used to generate a triggering signal for negative bitline voltage generator circuit. The bitline signal drops to zero level in case of the late arrival of the low voltage signal. Moreover, the variation of the delay in the delay chain should be larger than the bitline discharge rate of the write driver circuit for successful write operation. According to the detected variations between dummy line and cell bitline, a negative voltage is generated by the triggering circuit that mitigates the PVT variations.

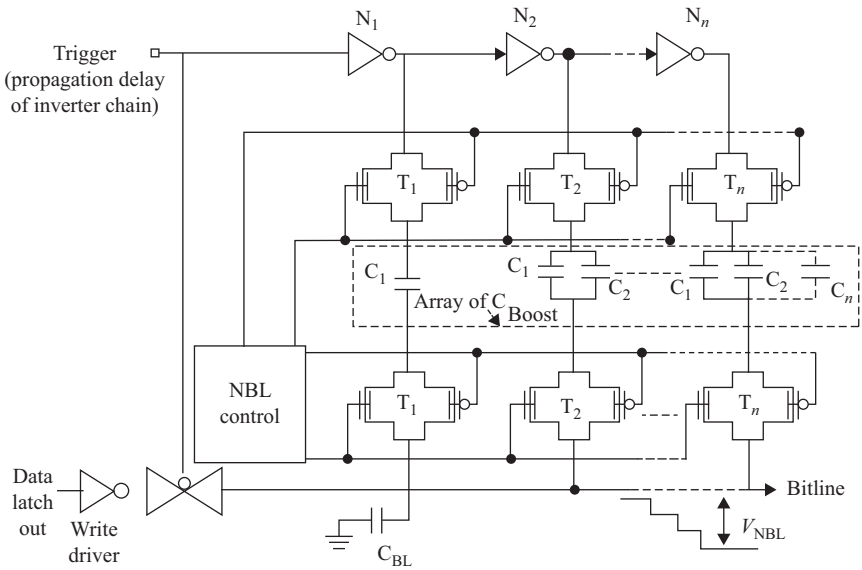


Figure 3.24 Trigger circuit for NBL-driven mitigation (Source: Dubey, 2014, p. 3 [60])

3.5.2.2 Leakage-driven design

Leakage-driven PVT-aware SRAM block schematic is shown in Figure 3.25. In this design, circuit-level mitigation technique detects PC of the die, and accordingly back-gate bias is applied to reduce the failure probability of IG-FinFET-based SRAM. The back-gate bias voltage is used to tune the device threshold voltage (V_{th}), and the SRAM array is designed using hybrid 10T SRAM cell (ref. hybrid 10T cell in Figure 3.26) [36,57]. This cell has pull-up and pull-down transistors for adjusting threshold voltage in order to achieve enhanced stability and performance. The cell ratio depends on pull-up and pull-down transistors sizing, and the back-gate bias of these transistors is sufficient for stability recovery [42].

The leakage-monitoring circuit is used to detect the process variations in SRAM, and the leakage current is used to identify the PCs. On the basis of the identified PCs, dynamic back-gate voltage generator circuit generates the appropriate back-gate bias voltage.

The selective back-gate voltages for nFinFET (V_{BGn}) and pFinFET (V_{BGp}) are generated based on the PCs detected by the sensor circuit. The back-gate voltage for mitigation with respect to the reference voltage depends on: (1) on-chip mechanism to detect PCs and (2) back-gate voltage applied to each transistor to amend SRAM failure probabilities and to enhance the performance. The variation in leakage current of SRAM array depends on device threshold voltage variation [57] and is detected by using a current mirror circuit [61] as shown in Figure 3.27(a). The measured leakage current is further amplified and stored in the latch. The output voltage of

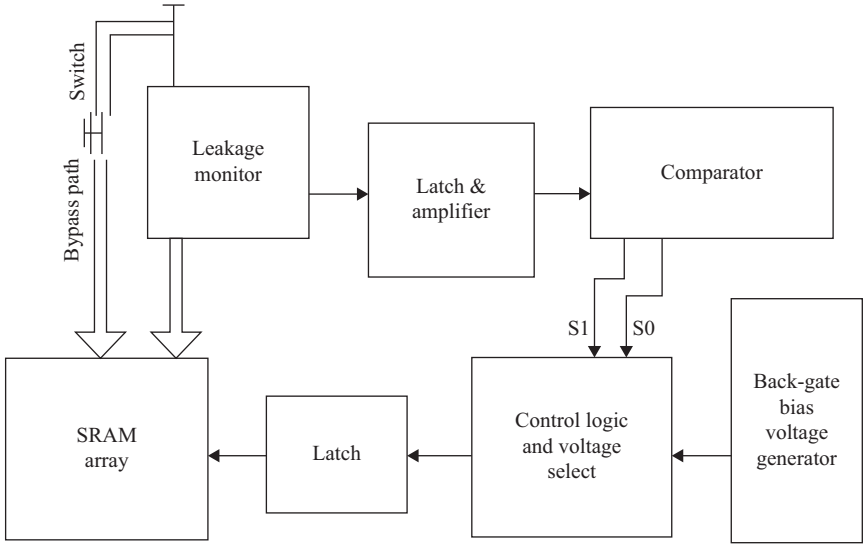


Figure 3.25 Block schematic of leakage-driven PVT-aware SRAM

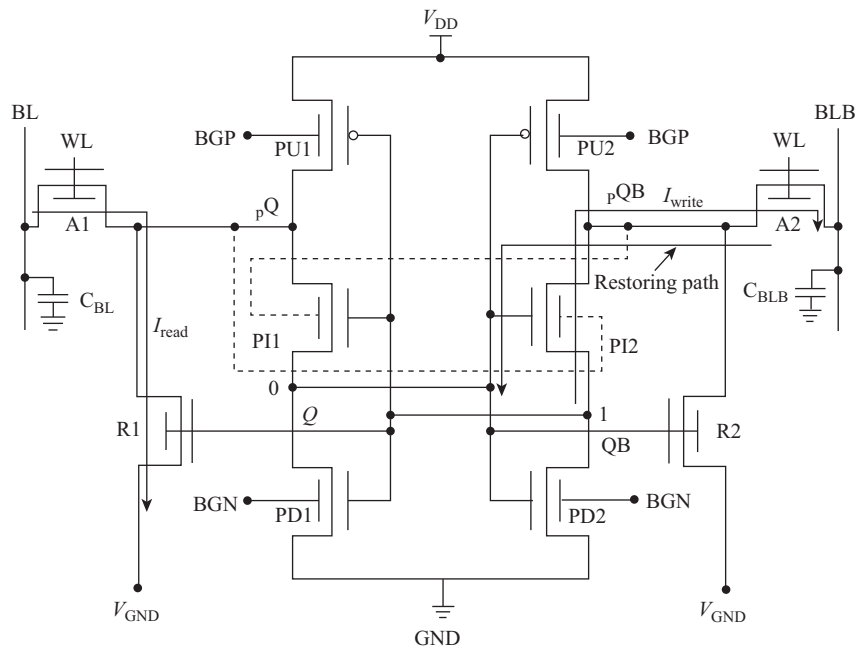


Figure 3.26 Hybrid 10T SRAM cell with selective back-gate biasing

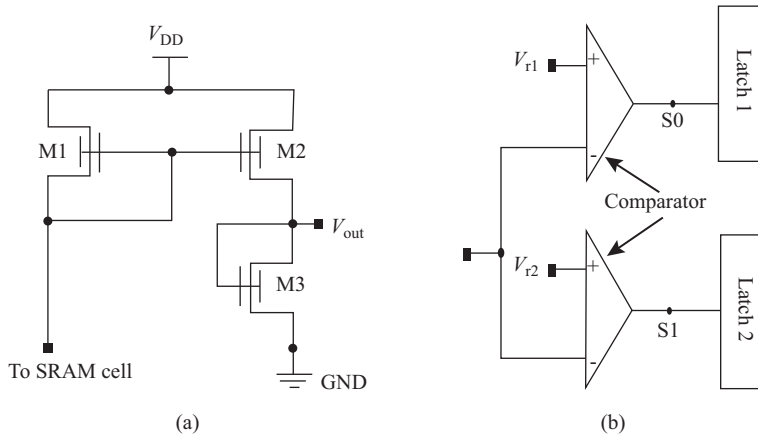


Figure 3.27 Peripheral circuits for leakage-driven PVT-aware SRAM. (a) Current mirror and (b) logic comparator

this stage is used to identify the PCs. Larger transistor sizes in the current mirror circuit are taken into account to reduce the effect of process variation. Further, it is assumed that the mirror circuit is not influenced by the process variation. The measured output voltage is then compared with the reference voltages using two parallel comparators as shown in Figure 3.27(b). Analog differential amplifiers are used to realize these comparators [57]. Based on the measured PCs, the back-gate voltage is selected from forward, reverse and zero voltage values and applied to back gates of the pull-up and pull-down transistors of the SRAM cell. A conventional multiplexer-based logic circuit can also be employed to select required voltages [5], and a latch circuit is used to store the selected back-gate bias voltage for SRAM array. Mostly, the mitigation is applied in standby modes of operation and in other modes of operation, bypass switch will be ON that provides a direct path from V_{DD} to the SRAM array. The PCs due to PVT variation are identified by the leakage current as discussed below.

Identification of PCs

Post-silicon SRAM dies fall into three categories: low threshold corner (Fast-Fast: FF), high threshold corner (Slow-Slow: SS) and nominal threshold corner (Typical-Typical: TT). The nominal threshold corner (TT) is a typical design value of threshold voltage for targeted technology.

As the leakage current shows exponential dependence on threshold voltage, small inter-die V_{th} variation results in large leakage shift of the SRAM cell. On the other hand, intra-die V_{th} variations cause more spread in Gaussian distribution plots of leakage current. Under large intra-die process variation, distribution plots are overlapping as shown in Figure 3.28(a). Therefore, identification of PCs is difficult under intra-die variation using single-cell leakage current monitoring. Further, the intra-die threshold voltage variations in different cells of the SRAM array are

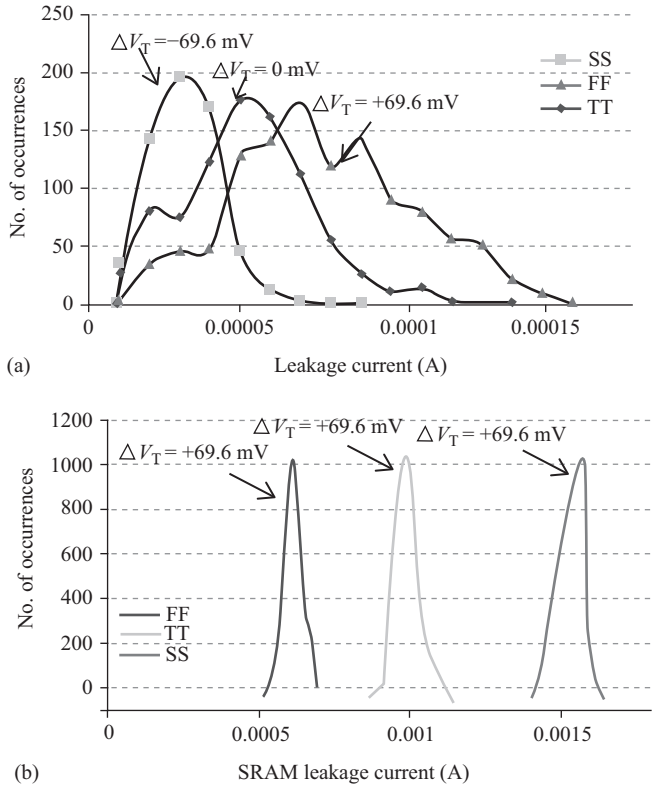


Figure 3.28 Effect of random intra-die V_{th} variation at different inter-die V_{th} PCs with leakage current distribution for (a) IG-FinFET SRAM cell and (b) IG-FinFET SRAM array

independent [2]. Thus, resulting curves are overlapping, and these can be separated using central limit theorem [62]. The central limit theorem states that the distribution of a random variable F (which is summation of large number of independent variables such as $A_1, A_2, \ldots, A_n$) is assumed normal with each variable mean (μ_F) given by:

$$\mu_F = \sum_{i=1}^{n} \mu A_i \tag{3.12}$$

where n is the number of variables. Moreover, the standard deviation (σ_F) is given by:

$$\sigma_F = \sum_{i=1}^{n} \sigma A_i \tag{3.13}$$

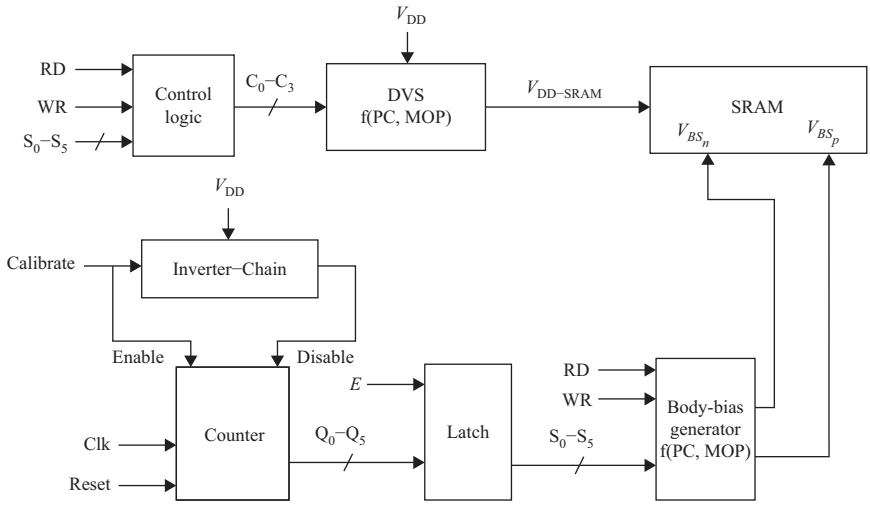


Figure 3.29 Block schematic of sensitivity-driven PVT-aware SRAM (Source: Yadav, 2014, p. 119 [63])

Symmetric mean and standard deviation of all the random variable are given by

$$\mu_M = N_{\text{cell}}\,\mu_{\text{cell}} \tag{3.14}$$

$$\sigma_M = \sqrt{N_{\text{cell}}\,\sigma_{\text{cell}}} \tag{3.15}$$

$$\frac{\mu_{\text{cell}}}{\sigma_{\text{cell}}} = \frac{1}{\sqrt{N_{\text{cell}}}}\frac{\sigma_M}{\mu_M} \tag{3.16}$$

where N_{cell} is the number of cells in a leakage-monitoring array (N_{cell} is a random variable for central limit theorem), μ_M and μ_{cell} are the mean values of SRAM array and cell, respectively, σ_M and σ_{cell} are the standard deviations of SRAM array and cell, respectively, from (3.16), it is clear that when number of random variable increases, the dispersion (standard deviation/mean) of the variable F reduces, *i.e.* adding more number of variables reduces random variation. The distribution curves for 512 cells show almost zero deviation in the threshold voltage due to intra-die variation for different PCs as shown in Figure 3.28(b). The detailed SRAM array size calculation is discussed in Reference 57. The limitation of this technique is that it mitigates only inter-die process variations and not exploiting the advantages of PCs in different MOPs.

3.5.2.3 Sensitivity-driven design

PVT-aware SRAM design using sensitivity-driven mitigation technique is shown in Figure 3.29. Sensitivity-driven PVT mitigation is a circuit-level technique that detects PC and MOP of the die and accordingly back-gate biasing or DVS is used to reduce

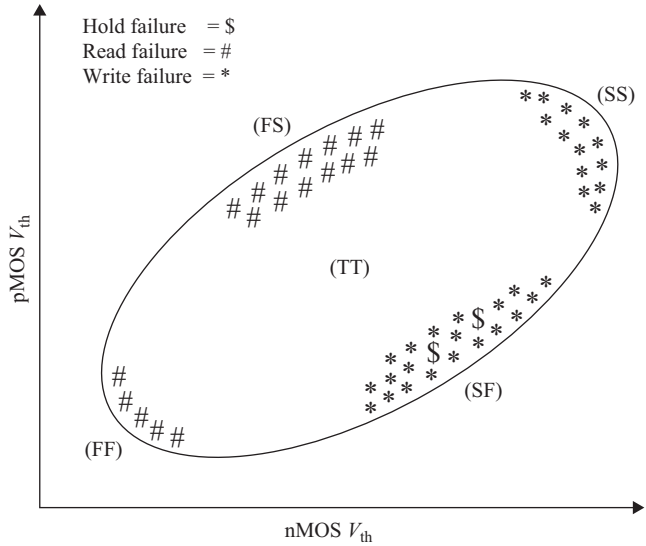


Figure 3.30 SRAM stabilities in different PCs

the failure probability. The back-gate bias voltage is used to tune the device threshold voltage (V_{th}), while DVS is used to tune the circuit performance such as delay and power by controlling the supply voltage (V_{DD}). Simulation results of SRAM under different PCs are shown in Figure 3.30. The hold failure occurs in SF PC, read failure occurs in FS and FF PCs, and write failure occurs in SS and SF PCs. Hence, based on the MOP and the detected PCs, back-gate biasing and DVS is applied such that it mitigates variations in SRAM.

The back-gate voltage and dynamic supply voltages are generated based on the MOPs, *i.e*. read, write and standby and the PCs, *i.e*. SS SF, TT, FS and FF. According to PC and MOP, the reverse back-gate-bias, forward back-gate bias or zero back-gate-bias (ZBB) is applied to amend SRAM failure probability and to limit the power budget. If the generated back-gate voltage is not sufficient to mitigate the large variation in the design, then DVS is applied that results in reducing the leakage/dynamic power consumption. An inverter chain is used to estimate PCs, because the delay of the inverter chain depends on threshold voltage. FinFET inverters are designed identical to SRAM cell inverters in order to mirror the effect of process variation. Further, central limit theorem is exploited to determine the depth of the inverter chain, and minimum of 830 inverters is required to identify the PCs that is also confirmed through MC simulation. A counter circuit is used to determine the PC as per the measured delay, and at the end of the count, final state (Q_0-Q_5) of the counter is sampled to latch by the enable (E) signal. The DVS block generates an output voltage ($V_{DD-SRAM}$) depending on the input signal states (C_0–C_3). Depending on the latch output (S_0–S_5) and the MOP, suitable back-gate-bias is applied to SRAM cells that results in decrease in write-failure probability. The stability of PVT-aware IG-FinFET-based SRAM designs is analysed in the following subsection.

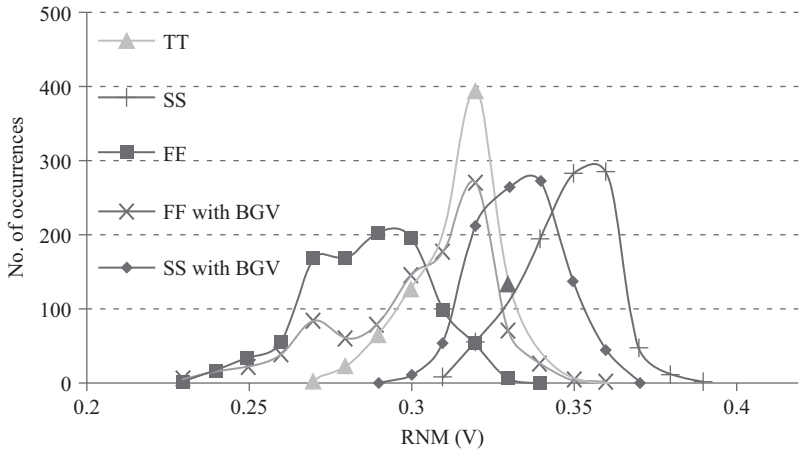


Figure 3.31 RNM distribution for leakage-driven PVT-aware SRAM

Table 3.1 Mean (μ) and deviation (σ) for RNM

S.No.	PC	TT	FF	SS	FF with BGV	SS with BGV
1	μ	0.31	0.28	0.34	0.3	0.32
2	σ	0.0138	0.018	0.014	0.02	0.013

3.5.3 Stability analysis

In order to analyse stability of PVT-aware SRAM designs, various simulations are run for leakage and sensitivity-driven IG-FinFET-based SRAM designs using 32 nm PTM technology [37]. These designs use hybrid-10 T SRAM cell (selective back-gate bias) to develop SRAM array.

Stability analysis for leakage-driven PVT-aware SRAM is first discussed with/without back-gate biasing. Figure 3.31 illustrates distribution of RNM for different PCs in the worst case using 5000 MC runs for with and without back-gate bias voltage (BGV) [63]. The corresponding mean value and standard deviation are shown in Table 3.1 [63]. The simulation results confirm that mean value is correct up to 94% and 96.7% for SS and FF PCs, respectively. The changes in standard deviation for FF and SS PCs positively are skewed by 0% and 31%, respectively, over without back-gate biasing. This analysis demonstrates that this technique delivers significant improvement in stability under process variation. Figure 3.32 shows read-failure probability for various PCs at different supply voltages. It further demonstrates that the built-in process tolerance in the back-gate bias hybrid 10 T cell gives lower read-failure probability. Also, along with RNM, WM is equally critical and shows write-ability of the cell, while WM of the SRAM cell shows ease of writing data in the SRAM cell.

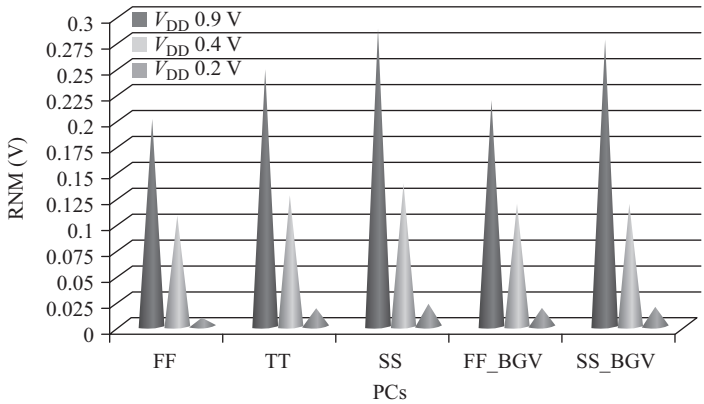


Figure 3.32 RNM versus PCs at different V_{DD} for leakage-driven SRAM

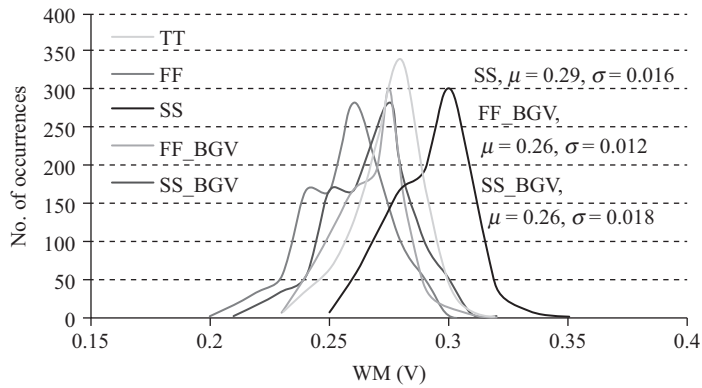


Figure 3.33 WM distribution for leakage-driven PVT-aware SRAM

Table 3.2 Mean (μ) and deviation (σ) for WM

S.No.	PC	TT	FF	SS	FF with BGV	SS with BGV
1	μ	0.27	0.25	0.29	0.26	0.26
2	σ	0.014	0.017	0.016	0.012	0.018

The WM distribution plots of write-failure probability for SRAM with and without back-gate bias voltage (BGV) are shown in Figure 3.33 [63]. The corresponding mean and standard deviation values are given in Table 3.2. The simulation results show up to 96% correction in mean value of RNM with standard deviation of 14% (negatively skewed) and 22% (positively skewed) at FF and SS PCs respectively. The positive back-gate bias increases the conductivity of the transistors and decreases

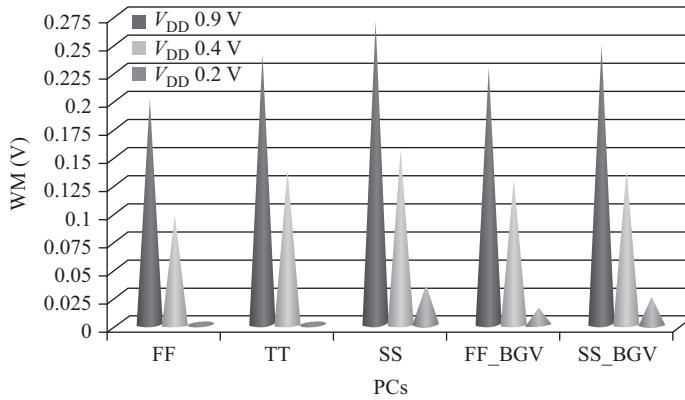


Figure 3.34 WM versus PCs at different V_{DD} for leakage-driven SRAM

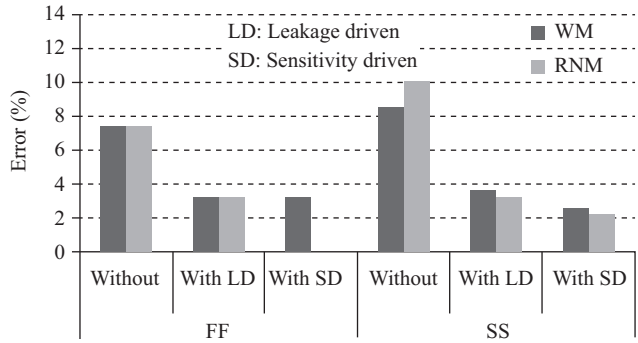


Figure 3.35 Percentage of error in RNM and WM for PVT-aware IG-FinFET SRAM designs

the inverter threshold voltage of the SRAM cell. This also increases the β-ratio in the cell due to the presence of back-gate conducting channel. Figure 3.34 shows write-failure probability for various PCs at different supply voltages. It further demonstrates that the built-in process tolerance in the back-gate bias hybrid 10T cell has lower write-failure probability. The stability analysis for leakage-driven PVT-aware SRAM is detailed in References 57 and 63.

As far as sensitivity-driven mitigation technique for PVT-aware SRAM design is concerned, it provides better improvement in RNM and WM as compared to the leakage-driven mitigation and is shown in Figure 3.35 [63]. The WM and RNM are measured for back-gate bias and without back-gate bias hybrid-10T SRAM cell at SS and FF PCs using spice simulation. The error in RNM and WM is calculated corresponding to detected PC for SRAM with and without back-gate bias voltage. The back-gate bias voltage is applied for recovery, and these results show that sensitivity-driven technique gives considerable improvement in recovery of the variations.

3.6 Conclusion

Due to device scaling in nanometre range, reliability has become a major challenge for IC designers, especially for future low-power, high-performance large SRAM designs. This fact motivated SRAM designers to use nanoscale FinFET devices and address the issues of PVT variations in designing reliable SRAMs. In this chapter, an attempt has been made to understand various FinFET device structures, SRAM topologies based on these FinFET devices and SRAM design challenges in the presence of PVT variations. In addition, recently proposed PVT-aware FinFET-based SRAM designs have been discussed in this chapter.

In nanoscale FinFET devices, the undoped fin is used to suppress the second-order effects such as subthreshold current, junction leakage and gate leakage. This thin fin improves mobility and switching speed of the device. Both bulk FinFET and SOI FinFET technology devices are discussed in this chapter. The SOI technology better controls the unwanted electric field effect resulting in reduced short-channel effect. The use of SOI material makes the fabrication of scaled devices easier, and therefore, SOI technology is commonly used for multi-gate device fabrication. However, the charge sharing at the corners of the fin is a major problem in multi-gate device, and this problem has been resolved in omega and independent-gate FinFETs. The IG-FinFET device is found as the most suitable device for designing stable and high-performance SRAMs. The use of back-gate biasing in various IG-FinFET-based SRAM topologies provides tolerance towards PVT variations. The back-gate bias IG-FinFET-based 6T and various PPN 10T SRAM topologies are presented along with their performance and stability analysis.

In order to consider the effects of PVT variations on stability and reliability of FinFET-based SRAMs in nanoscale region, PVT-aware SRAM design methodologies are discussed as a major contribution in this chapter. The SRAM designs employ static and dynamic mitigation techniques to mitigate the PVT variations. Both the techniques are explained along with their merits and demerits. Finally, dynamic PVT mitigation techniques namely NBL-, leakage- and sensitivity-driven techniques have been exploited in recently developed PVT-aware SRAM designs. The design details along with simulation results and analysis of all the proposed SRAM designs are given in this chapter. Based on the performance results of the analysed PVT-aware designs, it appears that sensitivity-driven IG-FinFET-based SRAM produces better results and may be more suitable for reliable and stable high-density memory designs. Moreover, the scope of this chapter may be extended to motivate the researchers towards developing device-circuit and circuit-system co-design PVT-aware approaches to provide robust and high-density memory designs for future generation multicore embedded systems in post-CMOS era.

References

[1] Mukhopadhyay S., Kim K., Mahmoodi H., Roy K. ‘Design of a process variation tolerant self-repairing SRAM for yield enhancement in nanoscaled CMOS’. *IEEE Journal of Solid-State Circuits*. 2007; 42(6): 1370–1382

[2] *International Technology Road Map for Semiconductors Report* [online]. 2011. Available from http://www.itrs.net/reports.html [Accessed 14 Jan 2013]
[3] Colinge J.-P. (eds.), *FinFETs and other multi-gate transistors*. Cambridge: Springer; 2008. pp. 9–28
[4] Chauhan Y.S., Venugopalan S., Paydavosi N., *et al*. 'BSIM compact MOSFET models for SPICE simulation'. *Proceedings of the 20th Mixed Design of Integrated Circuits and Systems (MIXDES) International Conference*; Jun. 2013. Gdynia: IEEE; 2013. pp. 23–28
[5] Suzuki K., Tosaka Y., Sugii T. 'Analytical threshold voltage model for short channel n+ − p+ double-gate SOI MOSFETs'. *IEEE Transactions on Electron Devices.* 1996; 43(5): 732–738
[6] Frank D., Laux S., Fischetti M. 'Monte Carlo simulation of a 30 nm dual-gate MOSFET: How short can Si go?'. *International Technical Digest on Electron Devices Meeting*; Dec. 1992. San Francisco, CA, USA: IEEE; 1992. pp. 553–556
[7] Hisamoto D., Kaga T., Kawamoto Y., Takeda E. 'A fully depleted lean-channel transistor (DELTA) – A novel vertical ultra thin SOI MOSFET'. *Electron Devices Meeting, International Technical Digest*; Dec. 1989. Washington, DC, USA: IEEE; 1989. pp. 833–836
[8] Hu C., Niknejad A., Sriramkumar V., *et al.* 'BSIM-IMG: A Turnkey compact model for fully depleted technologies'. *International SOI Conference*; Oct. 2012. NAPA, CA: IEEE; 2012. pp. 1–24
[9] Khandelwal S., Chauhan Y.S., Lu D.D., *et al.* 'BSIM-IMG: A compact model for ultrathin-body SOI MOSFETs with back-gate control'. *IEEE Transactions on Electron Devices.* 2012; 59(8): 2019–2026
[10] Nowak E., Ludwig T., Aller I., *et al*. 'Scaling beyond the 65 nm node with FinFET-DGCMOS'. *Proceedings of the Custom Integrated Circuits Conference*; Sep. 2003. San Jose, CA, USA; IEEE; 2003. pp. 339–342
[11] Park T.S., Cho H.J., Choe J.D., *et al*. 'Characteristics of body-tied triple-gate pMOSFETs'. *IEEE Electron Device Letters*. 2004; 25(12): 798–800
[12] Kim S.Y., Kim Y.M., Baek K.H., *et al*. 'Temperature dependence of substrate and drain–currents in bulk FINFETs'. *IEEE Transactions on Electron Devices.* 2007; 54(5): 1259–1264
[13] Ho B. 'Evolutionary MOSFET structure and channel design for nanoscale CMOS technology'. Doctoral thesis, University of California, Berkeley: 2012. pp. 24–25
[14] Anderson B.A., Bryant A., Nowak E.J. 'FinFET with low gate capacitance and low extrinsic resistance'. *Google Patents.* 2006
[15] Andrieu F., Dupré C., Rochette F., *et al.* '25 nm short and narrow strained FDSOI with TiN/HfO2 gate stack'. *Symposium on Digest of Technical Papers VLSI Technology*; Jun. 2006. Honolulu, HI: IEEE; 2006. pp. 134–135
[16] Burenkov A., Lorenz J. 'Corner effect in double and triple gate FinFETs'. *33rd European Solid-State Device Research Conference*; Sep. 2003. Estoril, Portugal: 2003. pp. 135–138
[17] Foster D. 'Subthreshold currents in CMOS transistors made on oxygen-implanted silicon'. *IEEE Electronics Letters*. 1983; 19(17): 684–685

[18] Bernard E., Ernst T., Guillaumot B., *et al.* internal spacers introduction in record high gate multichannel MOSFET satisfying both high-performance and low standby power requirements'. *IEEE Electron Device Letters.* 2009; 30(2): 148–151

[19] Ritzenthaler R., Dupré C., Mescot X., *et al.* 'Mobility behavior in narrow Ω-gateFETs devices'. *Proceeding of International SOI Conference*; Oct. 2006. Niagara Falls: IEEE; 2006. pp. 77–78

[20] Park T., Yoon E., Lee J.H. 'A 40 nm body-tied FinFET (OMEGA MOSFET) using bulk Si wafer'. *Physica E: Low-Dimensional Systems and Nanostructures*. 2003; 19(1):6–12

[21] Mishra P., Muttreja A., Jha N.K. 'FinFET circuit design'. *Nanoelectronic Circuit Design*. Springer; New York, NY, USA, 2011, pp. 23–54

[22] Yadav N., Dutt S., Pattanaik M., Sharma G. 'Double-gate FinFET process variation aware 10T SRAM cell topology design and analysis'. *European Circuit Theory and Design (ECCTD) Conference*; Sep. 2013. Dresden: IEEE; New York, NY, 10013, USA, 2013. pp. 1–4

[23] Hisamoto D., Lee W.C., Kedzierski J., *et al.* 'FinFET-a self-aligned double-gate MOSFET scalable to 20 nm'. *IEEE Transactions on Electron Devices.* 2000; 47(12): 2320–2325

[24] Lu D., Lin C.H., Niknejad A., Hu C. 'Compact modeling of variation in FinFET SRAM cells'. *Journal of Design and Test of Computers*. 2010; 27(2): 44–50

[25] Choi Y.K., King T.J., Hu C. 'A spacer patterning technology for nanoscale CMOS'. *IEEE Transactions on Electron Devices*. 2002; 49(3): 436–441

[26] Kang M., Song S., Woo S., *et al.* 'FinFET SRAM optimization with Fin thickness and surface orientation'. *Transactions on Electron Devices*. 2010; 57(11): 2785–2793

[27] Roy K., Mukhopadhyay S., Mahmoodi-Meimand H. 'Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits'. *Contributed Paper in Annual Proceedings of the IEEE*. 2003; 91(2): 305–327

[28] Li X., Parke S.A., Wilamowski B.M. 'Threshold voltage control for deep sub-micrometer fully depleted SOI MOSFET'. *Proceedings of the 15th Biennial* in *University/Government/Industry Microelectronics Symposium*; Jun. 2006. San Jose, CA: IEEE; 2003. pp. 284–287

[29] Sachid A.B., Francis R., Baghini M.S., *et al.* 'Sub-20 nm gate length FinFET design: Can high-κ spacers make a difference?'. Proceeding of *IEEE International* in *Electron Devices Meeting*; Dec. 2008. San Francisco: IEEE; 2008. pp. 1–4

[30] Raj B., Saxena A., Dasgupta S. 'Nanoscale FinFET based SRAM cell design: Analysis of performance metric, process variation, underlapped FinFET, and temperature effect'. *Circuits and Systems Magazine*. 2011; 11(3): 38–50

[31] Raj B., Mitra J., Bihani D.K., *et al.* 'Process variation tolerant FinFET based robust low power SRAM cell design at 32 nm technology'. *Journal of Low Power Electronics*. 2011; 7(2): 163–171

[32] Joshi R., Kim K., Kanj R. 'FinFET SRAM design'. *Nanoelectronic Circuit Design*; New York: Springer; 2011. pp. 55–95

[33] Carlson A., Guo Z., Balasubramanian S., *et al.* 'FinFET SRAM with enhanced read/write margins'. *Proceeding of International SOI Conference*; Oct. 2006. Niagara Falls, NY: IEEE; 2006. pp. 105–106

[34] Lo C.H., Huang S.Y. 'PPN based 10T SRAM cell for low-leakage and resilient subthreshold operation'. *Journal of Solid-State Circuits.* 2011; 46(3): 695–704

[35] Yadav N., Dutt S., Pattanaik M., Sharma G. 'Self-restoring PVT aware independently-controlled gate FinFET based 10T SRAM cell'. *Proceeding of the 25th International Conference on Microelectronics (ICM)*; Dec. 2013. Lebnon: IEEE; 2013, pp. 1–4

[36] Yadav N., Pattanaik M., Sharma G. 'New topology approach for future process, voltage and temperature aware SRAM using independently controlled double-gate FinFET'. *Journal of Low Power Electronics.* 2015; 11(1): 49–62

[37] *Predictive Technology Model* [online]. 2005. Available from http://ptm.asu.edu [Accessed [12 Jan. 2013]

[38] Ebrahimi B., Afzali-Kusha A., Mahmoodi H. 'Robust FinFET SRAM design based on dynamic back-gate voltage adjustment'. *Microelectronics Reliability*. 2014; (54)11: 2606–2612

[39] Burnett D., Parihar S., Ramamurthy H., Balasubramanian S. 'FinFET SRAM design challenges'. *Proceeding of International Conference on IC Design & Technology (ICICDT)*; May 2014. Austin, TX: IEEE; pp. 1–4

[40] Thakral G., Mohanty S.P., Ghai D. 'A combined DOE-ILP based power and read stability optimization in nano-CMOS SRAM'. *Proceeding of 23rd International Conference on VLSI Design,* India: IEEE; 2010. pp. 45–50

[41] Alam M. 'Reliability- and process-variation aware design of integrated circuits'. *Microelectronics Reliability*. 2008; 48(8–9): 1114–1122

[42] Grossar E., Stucchi M., Maex K., Dehaene W. 'Read stability and write-ability analysis of SRAM cells for nanometer technologies'. *IEEE Journal of Solid-State Circuits.* 2006; 41(11): 2577–2588

[43] Song T., Rim W., Jung J., *et al.* 'A 14nm FinFET 128Mb 6T SRAM with VMIN-enhancement techniques for low-power applications'. *Proceeding of Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*; Feb. 2014. IEEE; 2014, San Francisco, CA, pp. 232–233

[44] Fujimura Y., Hirabayashi O., Sasaki T., *et al.* 'A configurable SRAM with constant-negative-level write buffer for low-voltage operation with 0.149 μm 2 cell in 32 nm high-k metal-gate CMOS'. *Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*; Feb. 2010. San Francisco, CA; 2010. pp. 348–349

[45] Jan C.-H., Bhattacharya U., Brain R., *et al.* 'A 22nm SoC platform technology featuring 3-D tri-gate and high-k/metal gate, optimized for ultra low power, high performance and high density SoC applications'. *Proceeding of 2012 IEEE International* in *Electron Devices Meeting (IEDM)*; Dec. 2012. San Francisco, CA: IEEE; 2012. pp. 44–47

[46] Lee C.Y., Jha N.K. 'CACTI-FinFET: An integrated delay and power modeling framework for FinFET-based caches under process variations'. *Proceedings of the 48th Design Automation Conference*; Jun. 2011. New York, NY: 2011. pp. 866–871

[47] Mostafa H., Anis M., Elmasry M. 'Adaptive body bias for reducing the impacts of NBTI and process variations on 6T SRAM cells'. *Transactions on Circuits and Systems*. 2011; 52(12): 2859–2871

[48] Hsieh C.-Y., Fan M.-L., Hu V.-H., Su P., Chuang C.-T. 'Independently-controlled-gate FinFET Schmitt trigger sub-threshold SRAMs'. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems.* 2012; 20(7): 1201–1210

[49] Gupta V., Anis M. 'Statistical design of the 6T SRAM bit cell'. *IEEE Transactions on Circuits and Systems.* 2010; 57(1): 93–104

[50] Mukhopadhyay S., Mahmoodi H., Roy K. 'Reduction of parametric failures in sub-100-nm SRAM array using body bias'. *Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2008; 27(1): 174–183

[51] Datta A., Goel A., Cakici R.T., *et al.* 'Modeling and circuit synthesis for independently controlled double gate FinFET devices'. *Transactions on Computer-Aided Design of Integrated Circuits and* Systems. 2007; 26(11): 1957–1966

[52] Ebrahimi B., Rostami M., Afzali-Kusha A., Pedram M. 'Statistical design optimization of FinFET SRAM using back-gate voltage'. *Transactions on Very Large Scale Integration (VLSI) Systems*. 2010; 19(10): 1911–1916

[53] Thomas O., Reyboz M., Belleville M. 'Sub-1V, robust and compact 6T SRAM cell in double gate MOS technology'. *Proceeding of the International Symposium on Circuits and Systems, ISCAS*; May 2007. New Orleans, LA: IEEE; 2007. pp. 2778–2781

[54] Endo K., O'uchi S., Ishikawa Y., *et al.* 'Independent-double-gate FinFET SRAM for leakage current reduction'. *Electron Device Letters*. 2009; 30(7): 757–759

[55] Schuster S.E. 'Multiple word/bit line redundancy for semiconductor memories'. *Journal of Solid-State Circuits.* 1978; 13(5): 698–703

[56] Cao H., Liu M., Chen H., *et al.* 'Efficient built-in self-repair strategy for embedded SRAM with selectable redundancy'. *Proceeding of the 2nd International Conference on in Consumer Electronics, Communications and Networks (CECNet)*; Apr. 2012. Yichang, China: IEEE; 2012. pp. 2565–2568

[57] Mukhopadhyay S., Kang K., Mahmoodi H., Roy K. 'Reliable and self-repairing SRAM in nano-scale technologies using leakage and delay monitoring'. *Proceedings of International Test Conference*; Nov. 2005. Austin, TX: IEEE; 2005. pp. 1–10

[58] Mojumder N.N., Mukhopadhyay S., Kim J.J., *et al.* 'Design and analysis of a self-repairing SRAM with on-chip monitor and compensation circuitry'. *26th VLSI Test Symposium*; Apr–May 2008. San Diego, CA: 2008. pp. 101–106

[59] Goel A., Sharma R.K., Gupta A.K. 'Process variations aware area efficient negative bit-line voltage scheme for improving write ability of SRAM in nanometer technologies'. *Journal on Circuits, Devices & Systems.* 2012; 6(1): 45–51

[60] Dubey P., Ahuja G., Verma V., Yadav S.K., Khanuja A. 'A 500 mV to 1.0 V 128 Kb SRAM in Sub 20 nm Bulk-FinFET using auto-adjustable write assist'. *Proceeding of the 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems*; Jan. 2014, Mumbai: IEEE; 2014. pp. 150–155

[61] Marshall A., Kulkarni M., Campise M., *et al.* 'FinFET current mirror design and evaluation'. *Proceedings of the 2005 IEEE Dallas/CAS Workshop* in *Architecture, Circuits and Implementation of SOCs*; Oct. 2005. Richardson, TX: IEEE; 2005. pp. 187–190

[62] Papoulis A., Pillai S.U. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Company Limited; New Delhi, India, 2002. p. 308

[63] Yadav N., Dutt S., Sharma G. 'A New sensitivity-driven process variation aware self-repairing low-power SRAM design'. *Proceeding of 2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems*; Jan. 2014. Mumbai: IEEE; 2014. pp. 116–121

Chapter 4

Data stability and write ability enhancement techniques for FinFET SRAM circuits

Shairfe Muhammad Salahuddin[1] *and Volkan Kursun*[1]

Six-transistor static random-access memory (6T SRAM) cell is the fundamental building block of memory cache in modern microprocessors. Each bit of data is stored in an individual 6T SRAM cell in the memory subsystem. Read data stability and write ability of 6T SRAM cells are degraded with the scaling of CMOS technology. Conventional circuit techniques for achieving wider voltage margins during read and write operations cause significantly larger silicon area and increased power consumption. Several alternative FinFET memory design techniques are presented in this chapter for achieving stronger data stability during read operations and wider voltage margin during write operations without causing area and power consumption overheads in the memory subsystems of microprocessors.

4.1 Introduction

The capacity of on-die memory cache is increased to enhance the performance of modern microprocessors in each new CMOS technology generation [1–3]. Current industry standard static random-access memory (SRAM) cells are composed of six-transistors as shown in Figure 4.1. The data storage nodes (Node-1 and Node-2) of 6T SRAM cells are directly accessed by the bitlines during read operations [1–3]. The stored data is therefore disturbed. Memory banks are also important sources of leakage power consumption due to the large number of transistors in the SRAM arrays of modern microprocessors [1–3]. Achieving sufficient data stability, wide write voltage margin, and low leakage currents are the most important challenges in the design of nanoscale SRAM circuits. Novel SRAM cells are highly desirable for achieving more compact, robust, and energy-efficient memory circuits.

In this chapter, independent-gate bias, asymmetrical gate-to-source/drain underlap, and gate-underlap engineering are examined as FinFET technology enhancement options for achieving wider voltage margins during read and write operations in

[1]Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong

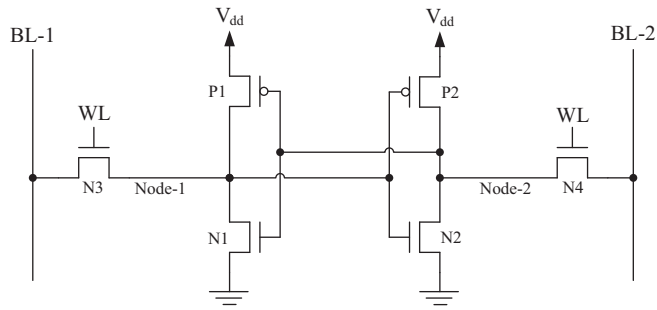


Figure 4.1 A standard SRAM cell with six transistors

memory circuits. This chapter is organized as follows. Six-FinFET SRAM cells are introduced in Section 4.2. SRAM cell layout and fabrication challenges are described in Section 4.3. As a case study, seven different 8Kbit memory arrays that are based on different SRAM cells and FinFET device structures are characterized and compared in Section 4.4. Impact of process parameter fluctuations on memory performance and reliability are investigated in Section 4.5. A summary of this chapter is given in Section 4.6.

4.2 Six-FinFET SRAM cells

SRAM cells that are composed of six FinFETs with different device structures are presented in this section. Conventional SRAM cell with symmetrically gate underlapped FinFETs is described in Section 4.2.1. An SRAM cell composed of independent-gate FinFETs (FinFET-Inde) is described in Section 4.2.2. An SRAM cell composed of asymmetrical FinFETs is introduced in Section 4.2.3. A hybrid SRAM cell with asymmetrical bitline access transistors is described in Section 4.2.4. A hybrid SRAM cell with asymmetrically gate-underlapped FinFETs is presented in Section 4.2.5. A single-ended read SRAM cell with underlap engineered FinFETs is described in Section 4.2.6.

4.2.1 Conventional six-FinFET SRAM cell

The conventional SRAM cell with six symmetrical FinFETs is presented in this section. The transistors that are used in this SRAM cell are symmetrically gate-to-source and gate-to-drain underlapped tied-gate FinFETs [4–7]. The symmetrically gate-underlapped FinFETs (FinFET-Sym) are designed and optimized to match the International Technology Roadmap for Semiconductors (ITRS) [8] projections for 15 nm FinFET technology node. The temperature for device simulation is 25°C [8]. Atlas 2D simulator [9] is used to characterize the SRAM cells. Quantum correction, carrier-carrier scattering, surface scattering, carrier velocity saturation, field-dependent mobility, concentration-dependent mobility, and temperature-dependent mobility

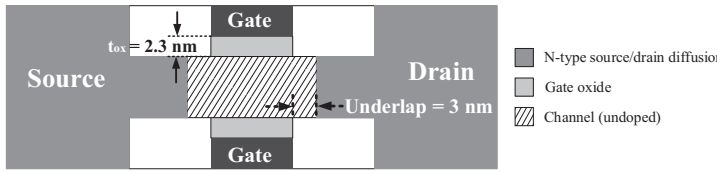


Figure 4.2 The cross-sectional view (along the gate length) of n-type tied-gate FinFET with symmetrical gate-underlaps (FinFET-Sym). Gate work-function = 4.4 6eV. Fin height = 15 nm

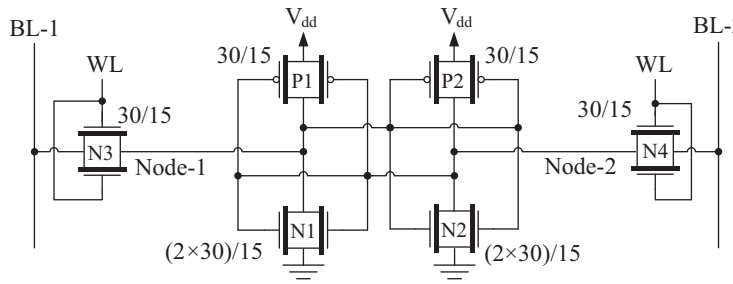


Figure 4.3 The conventional six-FinFET SRAM cell with symmetrically gate-underlapped tied-gate transistors (SRAM-Tied). All of the six transistors are tied-gate FinFETs with symmetrical gate-underlaps. The transistor sizes (width/length) are in nanometers assuming a 15 nm FinFET technology

models are used in the simulation. The gate length, fin thickness, and fin height of transistors are 15 nm, 6 nm, and 15 nm, respectively. The gate work-functions of the n-type and p-type FinFETs are 4.46 eV and 4.88 eV, respectively. Equivalent gate oxide thickness is 0.7 nm [8] (1.9 nm $HfO_2 + 0.4$ nm SiO_2). Gaussian source and drain doping profiles are assumed with peak doping concentration and characteristic length [9] of 2×10^{20} cm^{-3} and 2 nm, respectively. The profile of the optimized n-type FinFET-Sym is shown in Figure 4.2. The on-current and off-current that are produced by the minimum size (single-fin) n-type FinFET are 1756 μA/μm and 82 nA/μm, respectively.

An SRAM cell with symmetrical tied-gate FinFETs (SRAM-Tied) is shown in Figure 4.3 [1]. All the transistors in SRAM-Tied have identical gate underlaps. The bitline access transistors in SRAM cells are preferred to be weaker as compared to the pull-down transistors for lowering the disturbance of voltage on data storage nodes during read operations [1–3]. In order to maintain sufficient read data stability with SRAM-Tied, the β-ratio is assumed to be 2 [2] (two fins in each of the pull-down transistors) in this study. The bitline access transistors are preferred to be stronger as compared to the pull-up transistors in cross-coupled inverters for providing wider voltage margin and higher data transfer speed during write operations [1–3]. The pull-up transistors in SRAM-Tied are therefore designed to produce half of the on-current as compared to the bitline access transistors in this study.

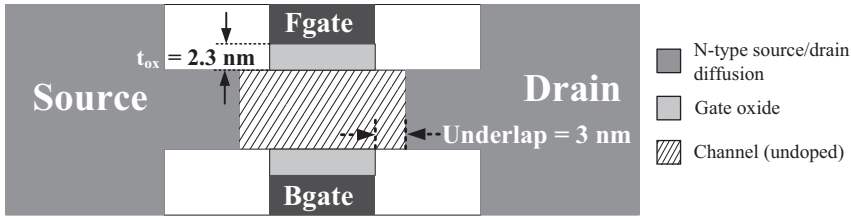


Figure 4.4 The cross-sectional view of n-type independent-gate FinFET (FinFET-Inde). Fgate: front-gate. Bgate: back-gate. Two gates are untied and independently biased/controlled

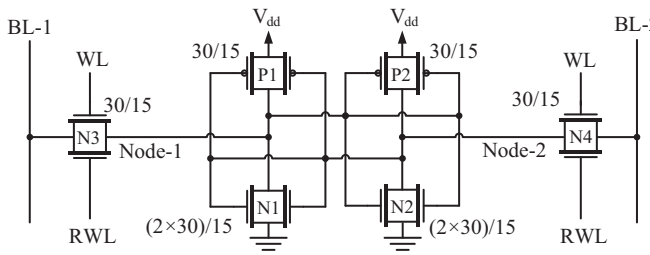


Figure 4.5 Six-FinFET symmetrical SRAM cell with independent-gate bitline access transistors (SRAM-Inde) [3]. WL, write wordline. RWL, read-write wordline. The transistor sizes (width/length) are in nanometers assuming a 15 nm FinFET technology

4.2.2 Independent-gate FinFET SRAM cell

A symmetrical SRAM cell with independent-gate bitline access transistors (SRAM-Inde) is proposed in Reference 3 for achieving enhanced data stability as compared to SRAM-Tied during read operations. Similar to SRAM-Tied, the cross-coupled inverters are composed of tied-gate symmetrical transistors in SRAM-Inde. The bitline access transistors are however independent-gate FinFETs. The independent-gate FinFET (FinFET-Inde) profile that is used in this study is shown in Figure 4.4. Underlap length and size of each of the transistors in SRAM-Inde are identical to SRAM-Tied. SRAM-Inde is shown in Figure 4.5.

In a FinFET-Inde, two vertical gates are separated by an oxide on top of the silicon fin [3, 10, 11], as shown in Figure 4.4. A FinFET-Inde operates in the dual-gate mode when both gates are biased to induce channel inversion. Alternatively, an n-type FinFET-Inde operates in the single-gate mode when one of the gates is deactivated by connection to ground. Disabling one of the gates in the single-gate mode reduces the on-current of a FinFET-Inde due to the weaker field-effect as compared to the dual-gate mode. The independent-gate bitline access transistors operate in the single-gate mode during read operations. The read data stability is thereby enhanced with SRAM-Inde as compared to SRAM-Tied. Alternatively, the bitline access transistors operate in the dual-gate mode during write operations. SRAM-Inde thereby provides a

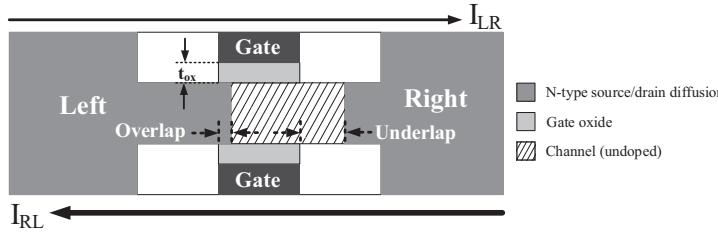


Figure 4.6 Cross-sectional view (along the gate length) of an asymmetrical n-type tied-gate FinFET-Asym1

write voltage margin that is similar to SRAM-Tied. The data access delay is however significantly increased due to weaker bitline access transistors during read operations with SRAM-Inde as compared to SRAM-Tied. The FinFET-Inde technology also increases the number of processing masks and fabrication complexity. Furthermore, the layout area is increased due to the additional gate-contacts in FinFET-Inde. Memory integration density (bits/μm^2) is thereby degraded with SRAM-Inde.

4.2.3 SRAM cell with asymmetrically overlap/underlap engineered FinFETs

The cross-sectional view of an n-type asymmetrically gate overlap/underlap engineered tied-gate FinFET (FinFET-Asym1 [12]) is shown in Figure 4.6. The left side of the channel is overlapped, while the right side is underlapped by the gate terminal in the asymmetrical FinFET-Asym1 structure [12].

The on-current that is produced by an asymmetrical FinFET varies depending on the direction of current flow. When the voltage level of the left electrode is higher than the right electrode, the right electrode is the source of charge carriers, while the left electrode is the drain terminal in an n-type FinFET-Asym1. The underlap region on the source side (right side) is far away from the gate. The fringing electric field that emerges from the gate cannot induce sufficient number of carriers in the underlap region on the source side. The channel resistance is therefore increased. Due to the asymmetrical design of FinFET-Asym1, the on-current flowing from the left to the right (I_{LR}) is reduced as compared to a symmetrical FinFET with identical gate length and channel width. Suppressing I_{LR} of bitline access transistors is desirable for achieving stronger data stability during read operations in SRAM cells.

Alternatively, when the voltage level of the right electrode is higher than the left electrode, the left electrode is the source of charge carriers while the right electrode is the drain terminal in an n-type FinFET-Asym1. Since the left side of the device is gate overlapped, a higher concentration of carriers is induced in the channel area with stronger field-effect. Drain induced depletion on the right side of the channel further reduces the channel resistance at the gate-underlap region. As illustrated in Figure 4.6, the on-current flowing from the right side of the device to the left side (I_{RL}) is significantly higher as compared to the left-to-right on-current (I_{LR}) due to the

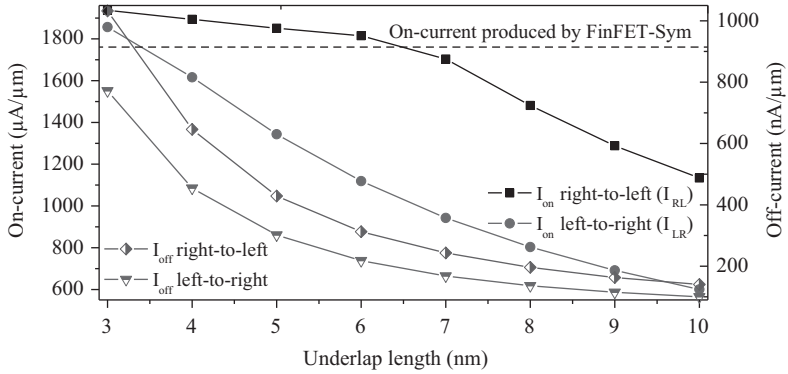


Figure 4.7 Variation of the on-current and off-current of n-type asymmetrical tied-gate FinFET-Asym1 with the underlap length. The underlap length on the right side is varied from 3 nm to 10 nm. The overlap on the left side is 1.5 nm. $T = 25°C$

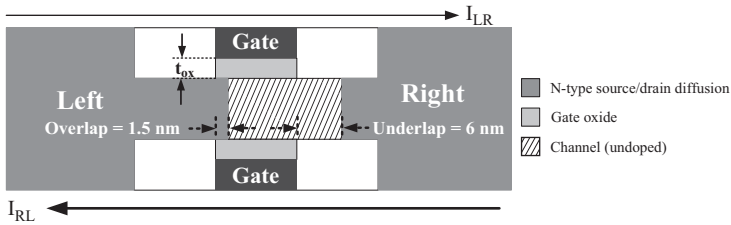


Figure 4.8 Cross-sectional view (along the gate length) of the asymmetrical n-type tied-gate FinFET-Asym1 that is used for SRAM cell design in this study. Gate work-function = 4.46 eV. Fin height = 15 nm. Fin thickness = 6 nm

asymmetrical design of FinFET-Asym1. Production of high I_{RL} by the bitline access transistors is critically important for achieving faster data transfer and wider voltage margin during write operations in 6T SRAM cells.

The gate overlap length is 10% of the nominal gate length in FinFET-Asym1 [1, 3, 12]. The underlap on the right side of the device is optimized for lowering I_{LR} as much as possible, while maintaining I_{RL} similar to the on-current of the optimum FinFET-Sym in this study. The right-to-left on-current (I_{RL}) is similar to the on-current of the optimum FinFET-Sym when the right underlap and left overlap lengths are 6 nm and 1.5 nm, respectively, as shown in Figure 4.7. The preferred underlap and overlap lengths in FinFET-Asym1 are therefore 6 nm and 1.5 nm, respectively. Due to shorter effective channel length in FinFET-Asym1, however, the off-currents that flow from right-to-left and left-to-right are increased by 3.8× and 2.7×, respectively, as compared to FinFET-Sym. The FinFET-Asym1 profile that is used for SRAM cell design in this study is shown in Figure 4.8. The saturation region drain current of FinFET-Asym1 and FinFET-Sym are compared in Figure 4.9 (a). I_{LR} that is produced by FinFET-Asym1 is 36.2% lower as compared to the on-current that is produced by FinFET-Sym.

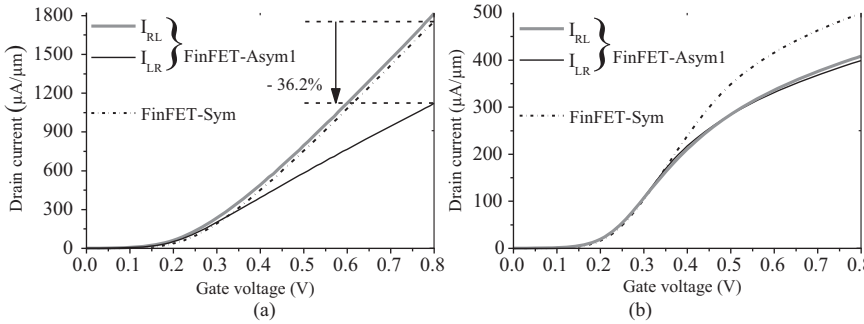


Figure 4.9 Comparison of drain currents that are produced by asymmetrical n-type tied-gate FinFET-Asym1 and symmetrical n-type tied-gate FinFET-Sym. (a) Saturation region drain current. $V_{drain} = 0.8\,V$. $V_{source} = 0\,V$. (b) Linear region drain current. $V_{drain} = 50\,mV$. $V_{source} = 0\,V$. $T = 25°C$

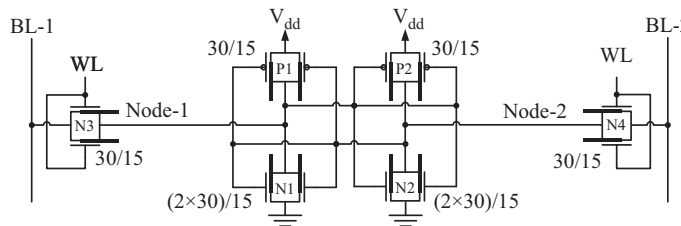


Figure 4.10 Purely asymmetrical SRAM cell (SRAM-Asym) [12]. All of the six transistors are tied-gate FinFETs with asymmetrical gate overlap/underlap. Asymmetrically gate overlap/underlap engineered FinFETs are represented with shifted asymmetrical thick lines along the channel. The transistor sizes (width/length) are in nanometers assuming a 15 nm FinFET technology

By employing asymmetrically gate overlap/underlap engineered FinFETs (FinFET-Asym1), a purely asymmetrical SRAM cell (SRAM-Asym) is proposed in Reference 12. The strength of bitline access transistors are automatically adjusted as the direction of current flow is reversed during read and write operations with this technique. SRAM-Asym is shown in Figure 4.10. All of the transistors in SRAM-Asym are tied-gate and asymmetrical (FinFET-Asym1). The gate-underlapped sides (right sides) of the pull-up, pull-down, and bitline access transistors are connected to the data storage nodes. The gate overlapped sides of the bitline access transistors are connected to the bitlines. The gate overlapped sides of the pull-up and pull-down transistors in cross-coupled inverters are connected to V_{dd} and ground, respectively.

The pull-down transistor that holds a "0" operates in linear region during read operations in a 6T SRAM cell. Higher linear region on-current is desirable in the pull-down transistors of cross-coupled inverters for providing stronger data stability during

read operations. The linear region drain currents of FinFET-Asym1 and FinFET-Sym are compared in Figure 4.9 (b). The linear region I_{LR} and I_{RL} that are produced by FinFET-Asym1 are 20.1% and 18.2% lower, respectively, as compared to the linear region on-current that is produced by FinFET-Sym. The data stability enhancement that is provided by SRAM-Asym is limited despite having weaker bitline access transistors as compared to the symmetrical memory cells. Furthermore, the leakage currents that are produced by the asymmetrical transistors in cross-coupled inverters are higher as compared to the symmetrical transistors. SRAM-Asym therefore consumes significant leakage power in idle mode.

4.2.4 Hybrid SRAM cell with asymmetrically overlapped/underlapped bitline access transistors

Symmetrical FinFETs are preferable in cross-coupled inverters for producing lower leakage currents in an SRAM cell. Alternatively, asymmetrical transistors are preferable for bitline access to achieve stronger data stability during read operations. Based on these observations, a hybrid SRAM cell (SRAM-Hybrid1) that employs both symmetrical and asymmetrical tied-gate transistors is proposed in Reference 13 as shown in Figure 4.11. Asymmetrically gate overlap/underlap engineered tied-gate FinFETs (FinFET-Asym1) are employed as bitline access transistors. Alternatively, cross-coupled inverters are implemented with symmetrical tied-gate FinFETs (see Figure 4.2 for the FinFET-Sym profile) in order to enhance read data stability and suppress subthreshold leakage currents with this technique as compared to the purely asymmetrical SRAM-Asym.

The operations of SRAM-Asym and SRAM-Hybrid1 are similar. The bitline access transistors are turned-on to initiate a read operation. The read current (I_{LR}) that flows from the bitline to the node that stores a "0" is weakened with FinFET-Asym1 as compared to a symmetrical transistor during read operations. Furthermore, the symmetrical pull-down transistor that stores a "0" in SRAM-Hybrid1 is stronger as compared to the asymmetrical pull-down transistors in SRAM-Asym during read

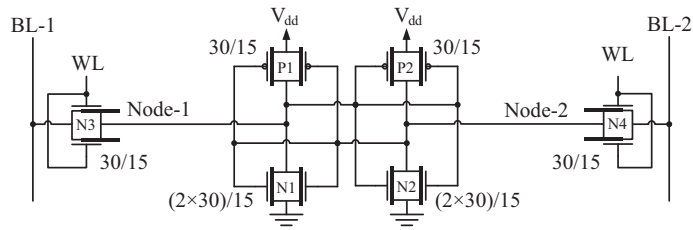


Figure 4.11 A hybrid six-FinFET SRAM cell (SRAM-Hybrid1). The bitline access transistors are asymmetrically gate overlapped/underlapped tied-gate FinFETs (FinFET-Asym1). The transistors in cross-coupled inverters are symmetrical tied-gate FinFETs (FinFET-Sym). The transistor sizes (width/length) are in nanometers assuming a 15 nm FinFET technology

operations. The voltage disturbances on data storage nodes are therefore smaller in SRAM-Hybrid1 as compared to both SRAM-Asym and SRAM-Tied. The read data stability is thereby enhanced with SRAM-Hybrid1 as compared to SRAM-Asym and SRAM-Tied.

The bitline access transistors are turned-on during write operations. Provided that a "0" is forced from BL-1 to Node-1 where a "1" was previously stored, the bitline access transistor (N3) competes with the pull-up transistor (P1) to discharge Node-1 from V_{dd} to 0 V. The gate overlapped terminal of N3 (the left side of the asymmetrical FinFET-Asym1 in Figure 4.8) that is connected to the bitline acts as the source of electrons. The channel series resistance of the asymmetrical bitline access transistor is reduced due to stronger field-effect and channel inversion at the source end. The write currents (I_{RL}) that flow from the data storage nodes to the bitlines in SRAM-Asym and SRAM-Hybrid1 are similar to SRAM-Tied. The pull-up transistor that holds a "1" is weaker in SRAM-Asym as compared to other SRAM cells. The write voltage margin of purely asymmetrical SRAM-Asym is therefore wider as compared to SRAM-Tied and SRAM-Hybrid1.

4.2.5 SRAM cell with asymmetrically gate-underlapped transistors

The hybrid asymmetrical FinFET SRAM cell (SRAM-Hybrid1) that is presented in Section 4.2.4 is effective in achieving stronger data stability and write ability. Due to shorter effective channel length, however, FinFET-Asym1 produces higher bitline leakage current as compared to symmetrical FinFETs in the idle mode. Larger bitline leakage currents increase the power consumption in idle mode. Furthermore, larger bitline leakage currents degrade the read speed and may cause read failure particularly at ultra-low power supply voltages [14]. An alternative hybrid FinFET SRAM cell with asymmetrically gate-underlap engineered tied-gate bitline access transistors that is proposed in Reference 15 is presented in this section to suppress the bitline leakage currents while providing similar read data stability and write ability as compared to SRAM-Hybrid1.

The cross-sectional view of an n-type asymmetrically gate-underlap engineered tied-gate FinFET (FinFET-Asym2) is shown in Figure 4.12. The underlap on the right

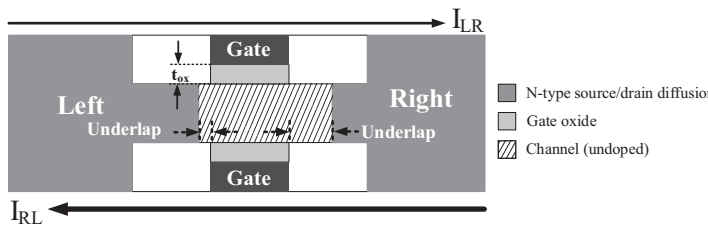


Figure 4.12 Cross-sectional view (along the gate length) of an asymmetrical n-type tied-gate FinFET-Asym2

side is longer as compared to the left side in an asymmetrical FinFET-Asym2 [16]. When the voltage level of the left terminal of FinFET-Asym2 is higher than the right terminal, the on-current flowing from the left side of the device to the right side (I_{LR}) is reduced due to the increased series resistance of the channel with the longer underlap region on the right side of the device. Suppressing the I_{LR} of bitline access transistors is desirable for providing stronger data stability during read operations. Alternatively, when the voltage level of the right electrode is higher than the left electrode, the on-current flowing from the right side of the device to the left side (I_{RL}) is increased due to the reduced resistance of the channel. As illustrated in Figure 4.12, I_{RL} is significantly larger as compared to I_{LR} due to the asymmetrical design of FinFET-Asym2. Maintaining high I_{RL} in bitline access transistors is critical for achieving faster data transfer and wider voltage margin during write operations.

The underlap length on the left side of FinFET-Asym2 is varied from 0 nm to 3 nm in this device optimization study. For each underlap length on the left side, the underlap length on the right side is increased from 3 nm to 10 nm for lowering I_{LR} as much as possible while maintaining I_{RL} similar to the on-current of FinFET-Sym. The right-to-left on-current (I_{RL}) is similar to the on-current of FinFET-Sym when the right underlap length is 6 nm with a left underlap of 2 nm in FinFET-Asym2, as shown in Figure 4.13. The preferred underlap lengths on the left and right sides are therefore 2 nm and 6 nm, respectively. Device profile of the FinFET-Asym2 that is used for SRAM cell design in this study is shown in Figure 4.14. Due to longer effective channel length, the left-to-right off-current that is produced by FinFET-Asym2 is 37.8% lower as compared to FinFET-Sym. Alternatively, the right-to-left off-current that is produced by FinFET-Asym2 is 27.9% lower as compared to FinFET-Sym, as shown in Figure 4.13. The saturation region currents that are produced by FinFET-Asym2 and FinFET-Sym are compared in Figure 4.15. The left-to-right on-current

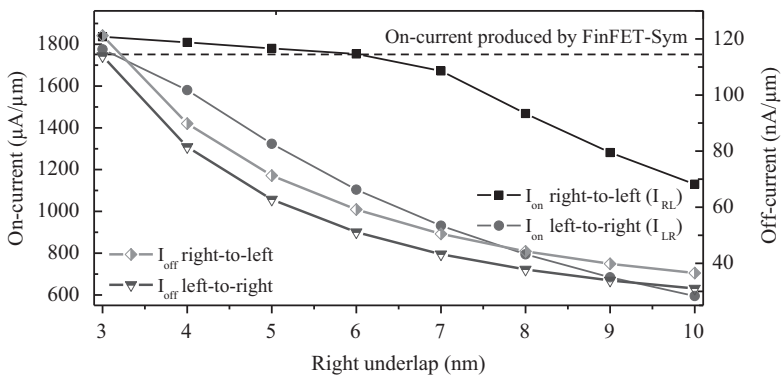


Figure 4.13 Variation of the on-current and off-current of n-type asymmetrical tied-gate FinFET-Asym2 with the underlap length. The underlap length on the left side of the gate is 2 nm. The underlap length on the right side of the gate is varied from 3 nm to 10 nm. T = 25°C

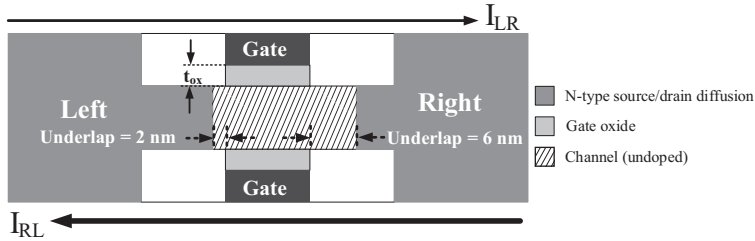


Figure 4.14 The cross-sectional view (along the gate length) of the asymmetrical n-type tied-gate FinFET-Asym2 that is used for SRAM cell design in this study. Gate work-function = 4.46 eV. Fin height = 15 nm. Fin thickness = 6 nm

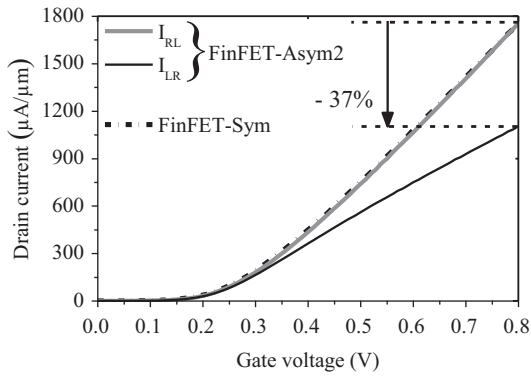


Figure 4.15 Comparison of saturation region drain currents that are produced by n-type FinFET-Asym2 and FinFET-Sym for various gate voltages. $V_{drain} = 0.8\,V$. $V_{source} = 0\,V$. $T = 25°C$

(I_{LR}) that is produced by FinFET-Asym2 is 37% lower as compared to FinFET-Sym in saturation mode as shown in Figure 4.15.

The low-leakage hybrid SRAM cell (SRAM-Hybrid2) that employs asymmetrically gate-underlap engineered tied-gate FinFETs (FinFET-Asym2) as the bitline access transistors, is shown in Figure 4.16. The shorter gate-underlapped sides are connected to the bitlines, while the longer gate-underlapped sides of the asymmetrical bitline access transistors (see Figure 4.14 for the FinFET-Asym2 profile) are connected to the data storage nodes. The cross-coupled inverters of SRAM-Hybrid2 are implemented with symmetrical tied-gate FinFETs (see Figure 4.2 for the FinFET-Sym profile).

The operation of SRAM-Hybrid2 is similar to SRAM-Asym and SRAM-Hybrid1. During read operations, the read current flows from the bitline to the

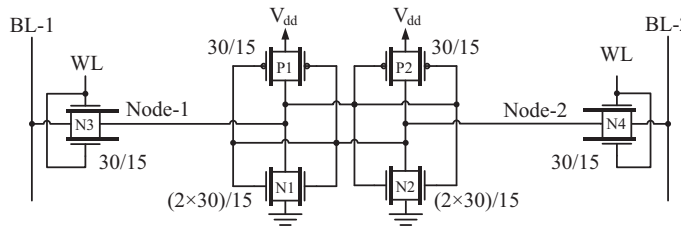


Figure 4.16 The second hybrid FinFET SRAM cell (SRAM-Hybrid2) with asymmetrically gate-underlapped tied-gate bitline access transistors. FinFET-Asym2s are represented with asymmetrical thick lines along the channel. The transistors in cross-coupled inverters are symmetrical tied-gate FinFETs (see Figure 4.2 for the FinFET-Sym profile). The transistor sizes (width/length) are in nanometers assuming a 15 nm FinFET technology

node that stores a "0" inside the memory cell. The bitline access transistor is weakened with higher channel resistance during read operations. The read data stability is thereby enhanced with the asymmetrically gate-underlap engineered bitline access transistors as compared to the SRAM cells with symmetrical bitline access transistors.

During write operations, the write current flows from the node that stores a "1" inside the memory cell to the discharged bitline. The write current is significantly higher than the read current in SRAM-Hybrid2. The transfer of new data into the SRAM cell is thereby facilitated with the asymmetrically gate-underlapped bitline access transistors. Furthermore, the bitline leakage currents that are produced by the hybrid asymmetrical SRAM-Hybrid2 are suppressed as compared to the purely asymmetrical SRAM-Asym, hybrid asymmetrical SRAM-Hybrid1, and symmetrical SRAM-Tied.

Fabrication of asymmetrically gate-underlapped single-gate MOSFETs are reported in References 17 and 18 by Advanced Micro Devices Incorporation and NEC Corporation, respectively. Prior to the source/drain doping, the gate spacer on the right and left sides of the devices are etched asymmetrically in References 17 and 18. Similar process can be used to fabricate asymmetrically gate-underlapped FinFETs. Furthermore, fabrication of symmetrically gate-underlapped transistors are reported in References 5, 19, 20, and 21. Prior to the source/drain doping, SiO_2 or SiN spacers are formed on both sides of the gate. The underlap length is controlled by the width of the gate spacer. Uniformity of the gate spacer is critical to avoid significant fluctuations in the electrical characteristics of devices. Uniform deposition of gate spacers is demonstrated with the fabrication of FinFETs with symmetrical gate-underlaps in Reference 5. Very-large-scale integration of symmetrically and asymmetrically gate-underlapped FinFETs for the realization of the hybrid memory circuits that are discuss in this chapter is therefore feasible.

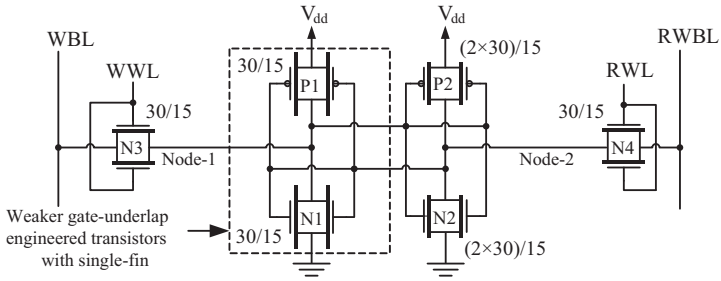


Figure 4.17 Single-ended read SRAM cell with underlap engineered symmetrical tied-gate FinFETs (SRAM-SR). The transistors in the left-side inverter are weaker as compared to the other transistors. Weaker gate-underlap engineered transistors are represented with longer thick lines along the channel. Underlap length in the left-side pull-down transistor = 14 nm. Underlap length in the left-side pull-up transistor = 4 nm. Underlap length of the other transistors = 3 nm. WWL: write wordline. RWL: read-write wordline. WBL: write bitline. RWBL: read-write bitline. The transistor sizes (width/length) are in nanometers assuming a 15 nm FinFET technology

4.2.6 Single-ended read SRAM cell with underlap engineered symmetrical-FinFETs

A single-ended read asymmetrical SRAM cell (SRAM-SR) with underlap engineered tied-gate symmetrical FinFETs is proposed in Reference 22 for achieving enhanced read data stability, stronger write ability, faster data transfer, and lower leakage power consumption in memory subsystems of microprocessors. SRAM-SR is shown in Figure 4.17. The underlap lengths of the transistors (N1 and P1) in the left-side inverter are longer as compared to the other transistors in SRAM-SR. Furthermore, the transistors in the left-side inverter of SRAM-SR are minimum sized (single fin). The bitline access transistors (N3 and N4) and the pull-down transistor (N2) in the right-side inverter of SRAM-SR are identical to SRAM-Tied. The underlap length of the pull-up transistor in the right-side inverter is identical to SRAM-Tied, SRAM-Inde, SRAM-Hybrid1, and SRAM-Hybrid2. However, the pull-up transistor in the right-side inverter of the single-ended read SRAM cell consists of two fins. The strengths of transistors in the left and right side inverters are therefore different. The voltage transfer characteristics (VTC) of the left and right side inverters in SRAM-SR are therefore not symmetrical, as shown in Figure 4.18.

The write operation with SRAM-SR is dual-ended. Prior to a write operation, depending on the incoming data, one of the bitlines of each accessed column of the memory array is discharged to 0 V. The WWL and RWL signals transition to V_{dd} to initiate the write operation. The bitline access transistors are turned on. New data and complementary data are forced into the SRAM cell from WBL and RWBL, respectively. Each of the transistors in the left-side inverter of SRAM-SR are minimum

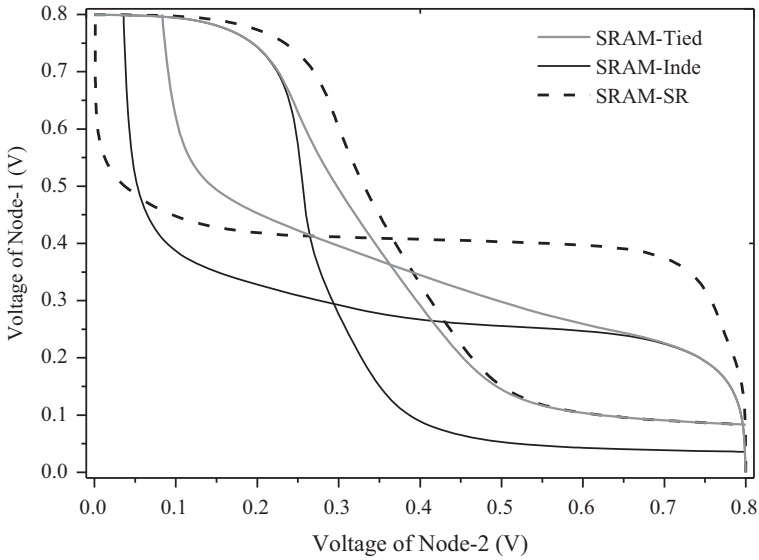


Figure 4.18 Butterfly curves of SRAM-Tied, SRAM-Inde, and SRAM-SR during read operations. $V_{dd} = 0.8\,V.\ T = 90°C$

sized (single-fin). The gate-underlap lengths of pull-up (P1) and pull-down (N1) transistors in the left-side inverter of SRAM-SR are also elongated to further weaken these two minimum sized transistors. The voltage margin and data transfer speed during write operations are therefore enhanced with SRAM-SR as compared to the other SRAM cells that are presented in this chapter. Furthermore, transistors with longer underlaps produce lower sub-threshold leakage currents. In the idle mode (WWL = RWL = 0 V), SRAM-SR consumes the lowest leakage power among all the memory cells that are presented in this chapter.

The read operation with SRAM-SR is single-ended. The read bitline is pre-charged to V_{dd} prior to a read operation. RWL transitions to V_{dd} to initiate the read operation. The right-side bitline access transistor (N4) is turned-on. Provided that a "0" is stored on Node-2, RWBL is discharged through the transistor stack that is formed by N4 and N2. Alternatively, if a "1" is stored on Node-2, the voltage level of RWBL is maintained at V_{dd}. Depending on the voltage level of RWBL, the stored data is detected by a sense amplifier that is attached to RWBL.

P2 and N2 consist of two fins for suppressing the data disturbance during read operations in SRAM-SR. The VTC of the right-side inverter in SRAM-SR is less skewed as compared to the other SRAM cells. The underlap lengths of the transistors (N1 and P1) in the left-side inverter of SRAM-SR are tuned to enhance the gain of the VTC in the transition region. The openings (eyes) of the butterfly curves of cross-coupled inverters are therefore wider with SRAM-SR as compared to the other SRAM cells (SRAM-Tied and SRAM-Inde are shown for comparison in Figure 4.18).

The single-ended read SRAM cell thereby provides stronger data stability during read operations as compared to the other SRAM cells that are presented in this chapter.

4.3 Fabrication and SRAM cell area comparison

For achieving superior electrical characteristics, the fin thickness of FinFETs is required to be less than half of the gate length [23, 24]. Conventionally, minimum feature size in integrated circuits is limited by the gate length. Fabricating FinFETs with conventional lithography technologies is therefore extremely challenging. A spacer based technology is proposed in Reference 23 for fabricating ultra-thin body FinFETs. In Reference 23, active fins are patterned using spacer layers that are deposited around sacrificial layers. The spacer based FinFET technology doubles the fin density as compared to the conventional lithography techniques. Spacer based FinFET fabrication technology [23] is assumed for maximizing the memory integration density in this chapter.

In a spacer based FinFET fabrication technology, fins are fabricated in groups. Two fins belong to a group. Unwanted fins are selectively removed from a circuit [2], as illustrated in Figure 4.19. The layout area occupied by a dual-fin transistor is therefore identical to a single-fin transistor. Conventionally, one fin from each of the pull-up transistors (in cross-coupled inverters) is selectively etched away [2] for maintaining wide voltage margin during write operations with SRAM cells. The pull-up transistor in the left-side inverter of SRAM-SR is minimum-sized (single-fin). Alternatively, the pull-up transistor in the right-side inverter has dual-fins. The additional fin in the right-side pull-up transistor of SRAM-SR does not cause any area overhead as compared to SRAM-Tied, SRAM-Asym, SRAM-Hybrid1, and SRAM-Hybrid2.

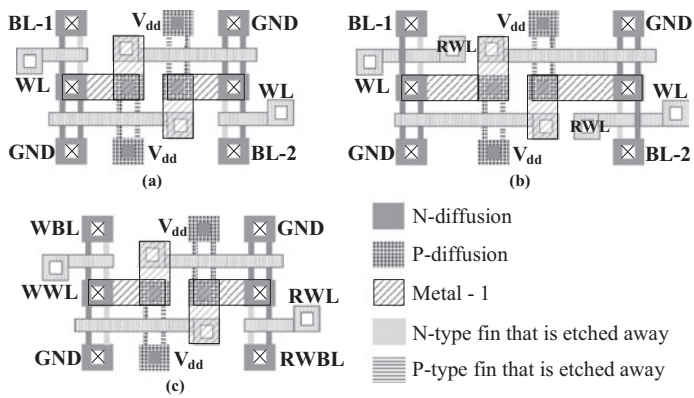


Figure 4.19 Layouts of SRAM cells in a 15 nm FinFET technology. (a) SRAM-Tied, SRAM-Asym, SRAM-Hybrid1, SRAM-Hybrid2. Cell area = 0.04365 μm^2. (b) SRAM-Inde. Cell area = 0.05715 μm^2. (c) SRAM-SR. Cell area = 0.04365 μm^2

Fabrication of gate-underlapped transistors are reported in References 13–16. Prior to the source/drain doping, sidewall spacers are formed on both sides of the gate (in source/drain extension regions). The underlap lengths are controlled by the width of the sidewall spacers. After source/drain ion implantation, device is covered by dielectric for further processing. Sidewall spacers do not increase the length of source/drain extensions as long as the underlap length is shorter than the extension regions. Underlap lengths of all the transistors are shorter than the source/drain extensions in this study. The lengths of the source and drain extensions in different FinFETs are therefore identical. Irrespective to the underlap lengths, the device footprints of all SRAM-Tied are equal in this study. The layout areas of SRAM-Tied, SRAM-Asym, SRAM-Hybrid1, SRAM-Hybrid2, and SRAM-SR are therefore identical. Due to the additional gate contacts, the layout area of SRAM-Inde is ∼31% larger as compared to the other SRAM cells, as shown in Figure 4.19.

4.4 Case study: 8Kbit memory arrays designed with different SRAM cells

The electrical characteristics of 8Kbit memory arrays with SRAM-Tied, SRAM-Inde, SRAM-Asym, SRAM-Hybrid1, SRAM-Hybrid2, and SRAM-SR memory cell structures are compared in this section assuming a 15 nm FinFET technology. The read data stability, write voltage margin, data access (read and write) delays, and leakage power consumption are evaluated in Sections 4.4.1–4.4.5. The nominal power supply voltage (V_{dd}) is 0.8 V and simulation temperature is 90°C [25] in this characterization.

4.4.1 Read static noise margin

Read static noise margins (RSNMs) of memory cells are characterized in this section. RSNM is the maximum DC voltage noise that can be tolerated by an SRAM cell without losing the data during read operations [14, 26–29]. The worst-case (narrowest) SNMs are observed at high temperature (90°C) in all the memory cells.

Variation of the read data stability of SRAM-SR with the gate-underlap lengths of the pull-up and pull-down transistors in the left-side inverter is shown in Figure 4.20. Data stability is enhanced (as shown in Figure 4.20) with the elongated underlap length of the left-side pull-down transistor until the two eyes of VTCs of cross-coupled inverters become approximately equal. Further increase in underlap length of the pull-down transistor in the left-side inverter degrades the data stability as the eyes of VTCs become imbalanced. The optimum gate-underlap lengths of the pull-up and pull-down transistors in the left-side inverter are 4 nm and 14 nm, respectively, for providing maximum read data stability with SRAM-SR. The RSNMs of FinFET SRAM cells are shown in Figure 4.21. SRAM-SR provides 76.2%, 66.1%, 38.5%, 37.9%, and 19.4% stronger read data stability as compared to SRAM-Tied, SRAM-Asym, SRAM-Hybrid1, SRAM-Hybrid2, and SRAM-Inde, respectively.

Due to weaker bitline access transistors during read operations, SRAM-Inde provides 47.6%, 39.1%, 16%, and 15.5% wider voltage margin during read operations

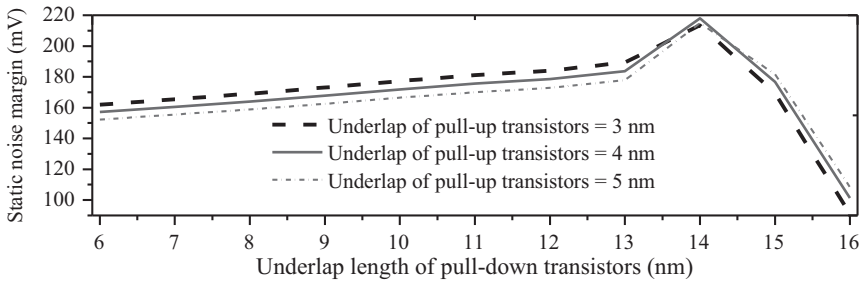


Figure 4.20 Variation of the data stability with the gate-underlap lengths of pull-up and pull-down transistors in the left-side inverter of SRAM-SR. $V_{dd} = 0.8\,V$. $T = 90°C$

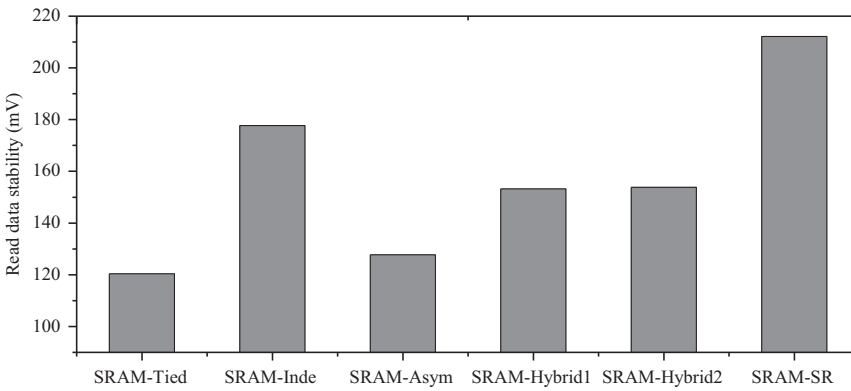


Figure 4.21 The read static noise margins of FinFET SRAM cells. $T = 90°C$

as compared to SRAM-Tied, SRAM-Asym, SRAM-Hybrid1, and SRAM-Hybrid2, respectively. The read currents (I_{LR}) that are produced by the bitline access transistors (FinFET-Asym2 and FinFET-Asym1) in SRAM-Hybrid2 and SRAM-Hybrid1 are lower as compared to SRAM-Tied. Furthermore, the symmetrical pull-down transistors (FinFET-Sym) in cross-coupled inverters of SRAM-Hybrid2 and SRAM-Hybrid1 have larger linear region on-currents as compared to the asymmetrical transistors in SRAM-Asym. The RSNMs of hybrid asymmetrical SRAM cells are enhanced by 27.7% and 20.4% as compared to SRAM-Tied and purely asymmetrical SRAM-Asym, respectively. SRAM-Hybrid1 and SRAM-Hybrid2 have similar data stability (difference less than 1%).

4.4.2 Hold static noise margin

The hold static noise margin (HSNM) of an SRAM cell is determined by the VTC of cross-coupled inverters in the idle mode. The HSNM of different FinFET SRAM

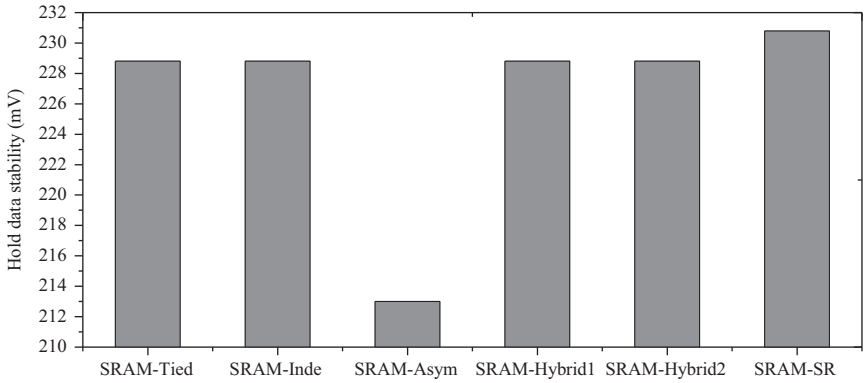


Figure 4.22 The hold static noise margins of FinFET SRAM cells. $T = 90°C$

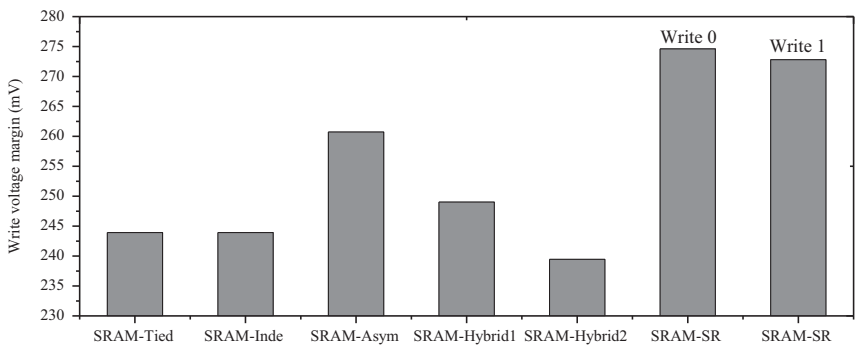


Figure 4.23 The write voltage margins of FinFET SRAM cells. $T = 90°C$

cells are shown in Figure 4.22. The hold data stability with SRAM-SR is enhanced by up to 8.4% as compared to the other SRAM cells.

FinFET-Sym has higher threshold voltage as compared to FinFET-Asym1. The VTC of cross-coupled inverters therefore have narrower transition regions with stronger output gain in memory cells that employ symmetrical transistors as compared to the memory cells with asymmetrical transistors. SRAM-Tied, SRAM-Inde, SRAM-Hybrid1, and SRAM-Hybrid2 thereby enhance HSNM by 7.4% as compared to SRAM-Asym.

4.4.3 Write voltage margin

The write voltage margins of SRAM cells are evaluated in this section. The write voltage margin is measured as described in Reference 30. The write voltage margins of FinFET SRAM cells are shown in Figure 4.23. Transistors in the left-side inverter of SRAM-SR are weaker as compared to the other SRAM cells that are investigated

in this study. The write operation is therefore facilitated with SRAM-SR. SRAM-SR enhances the voltage margin during write operations by up to (depending on the incoming data) 14.7% as compared to other memory cells.

The asymmetrical pull-up transistor that is holding a "1" in SRAM-Asym produces lower contention current as compared to SRAM-Tied, SRAM-Inde, SRAM-Hybrid1, and SRAM-Hybrid2. Write operations are thereby facilitated with SRAM-Asym. The write voltage margin of SRAM-Asym is by up to 8.8% wider as compared to SRAM-Tied, SRAM-Inde, SRAM-Hybrid1, and SRAM-Hybrid2. SRAM-Tied, SRAM-Inde, and the hybrid asymmetrical SRAM cells (SRAM-Hybrid1 and SRAM-Hybrid2) have similar (difference less than 5.1%) write voltage margins.

4.4.4 Data access speed

The worst-case (longest) read and write delays of the 128 × 64-bit memory arrays with different SRAM cells are compared in this section. The diffusion and gate capacitors of FinFETs are extracted from the SRAM cell layouts using the Atlas device simulator [9]. Bitline and wordline parasitic impedances are extracted with Clever [9]. Π-type RC networks are used to characterize the worst-case (longest) data access delays.

The read delay of an SRAM cell is the time interval from the 50% point of the WL low-to-high transition until a 200 mV voltage difference is developed between the bitlines. Due to stronger bitline access transistors, the read delays with SRAM-Tied and SRAM-SR are reduced by up to 55.3% as compared to other SRAM cells, as shown in Figure 4.24. Due to the independent-gate bitline access transistors, the read delay of SRAM-Inde is by up to 65.7% longer as compared to SRAM-Asym, SRAM-Hybrid1, and SRAM-Hybrid2. SRAM-Asym, SRAM-Hybrid1, and SRAM-Hybrid2 offer similar (difference less than 5.9%) read speed.

The write delay of an SRAM cell is the time interval from the 50% point of the WL low-to-high transition until one of the data storage nodes is charged from 0 V to $V_{dd}/2$. Due to the weaker transistors in the left-side inverter, write operation with

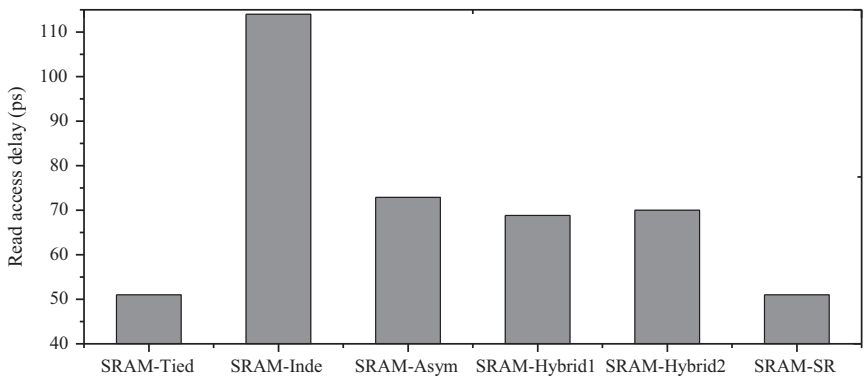


Figure 4.24 The read access delays of FinFET SRAM cells. $T = 90°C$

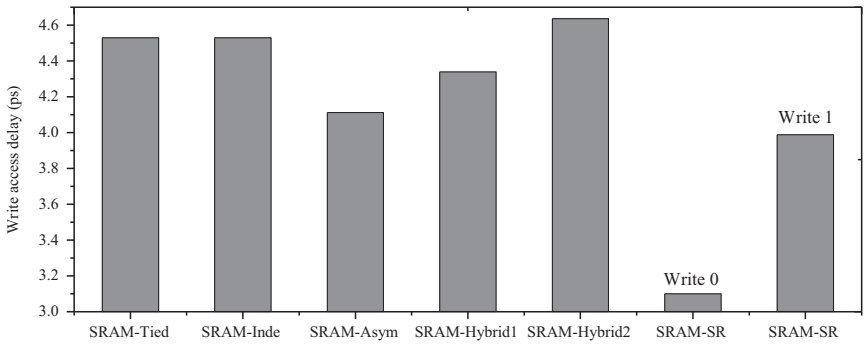


Figure 4.25 The write access delays of FinFET SRAM cells. $T = 90°C$

SRAM-SR is up to (depending on the incoming data) 33.1% faster as compared to the other SRAM cells.

During write operations, the pull-up transistors in cross-coupled inverters of the purely asymmetrical SRAM-Asym produce lower contention current as compared to SRAM-Tied, SRAM-Inde, SRAM-Hybrid1, and SRAM-Hybrid2. The worst-case write operation with SRAM-Asym is therefore up to 11.3%, 9.2%, and 5.2% faster as compared to SRAM-Hybrid2, SRAM-Tied/SRAM-Inde, and SRAM-Hybrid1, respectively, as shown in Figure 4.25. SRAM-Hybrid1, SRAM-Hybrid2, SRAM-Tied, and SRAM-Inde have similar write access delays (difference less than 4.4%).

4.4.5 Leakage power consumption

The leakage power consumed by different SRAM cells is compared in this section. The leakage currents are measured at 90°C (assuming a short idle period near a hot spot). The leakage power consumptions of SRAM cells are shown in Figure 4.26. Due to weaker transistors in the left-side inverter, SRAM-SR reduces the leakage power consumption by up to 73.7% as compared to the other SRAM cells.

Due to longer effective channel length, the asymmetrical FinFET-Asym2 produces lower leakage currents as compared to the symmetrical FinFETs that are used in SRAM-Tied and asymmetrical FinFETs that are used in SRAM-Asym1/SRAM-Hybrid1. The leakage power consumption of SRAM-Hybrid2 is therefore reduced by 65.6%, 34.6%, and 10.5% as compared to SRAM-Asym, SRAM-Hybrid1, and SRAM-Tied/SRAM-Inde, respectively.

While offering significantly stronger data stability and higher data access speed, SRAM-SR also consumes lower leakage power as compared to the other SRAM cells that are investigated in this chapter. SRAM-SR is therefore attractive for achieving robust, low-power, and high-performance memory sub-systems in modern microprocessors.

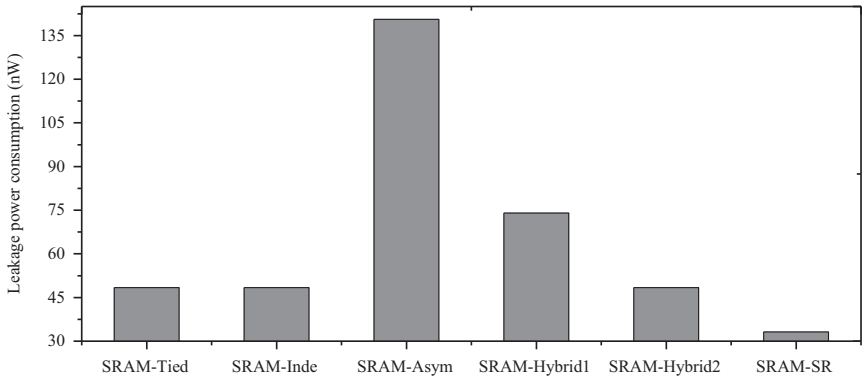


Figure 4.26 The leakage power consumptions of SRAM cells in idle mode (WL = RWL = 0V). T = 90°C

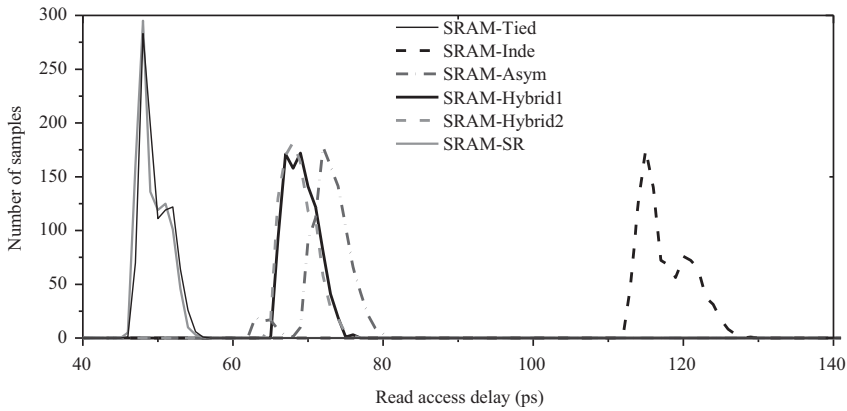


Figure 4.27 Statistical read access delay distributions of FinFET SRAM cells. T = 90°C

4.5 Variations of underlap (overlap) lengths due to process imperfections

The electrical characteristics of SRAM cells become increasingly sensitive to process parameter fluctuations with CMOS technology scaling [1, 3]. Memory cells are evaluated under gate-underlap (overlap) length variations in this section. 1000 Monte-Carlo simulations are run with Atlas. The left-side underlap (overlap) and right-side underlap lengths are assumed to have independent Gaussian distributions. The parameters of each transistor are varied independently. The underlap (overlap) lengths of FinFETs have 3σ variations of 2 nm. The statistical distributions of read access delay, leakage power consumption, write voltage margin, and RSNM of different SRAM cells are shown in Figures 4.27–4.30.

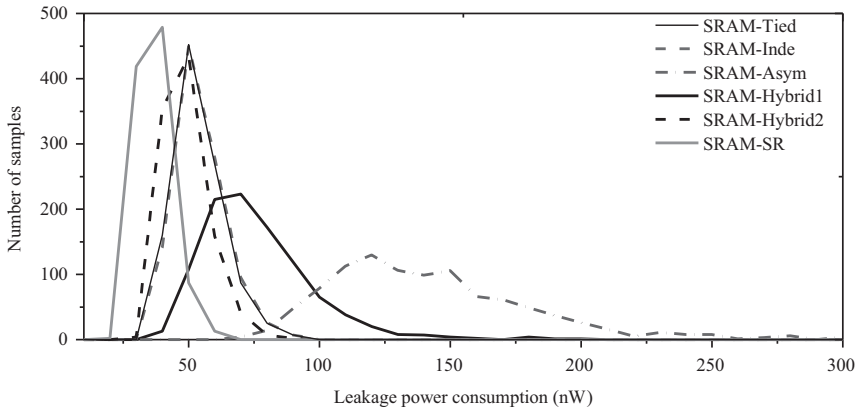


Figure 4.28 Statistical leakage power consumption distributions of FinFET SRAM cells. $T = 90°C$

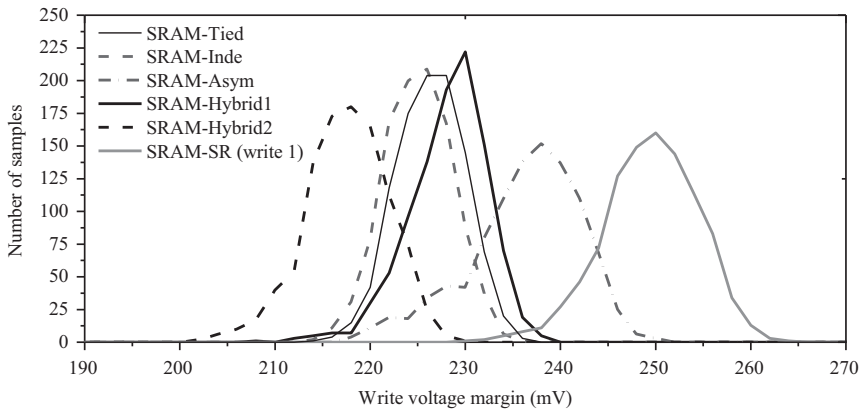


Figure 4.29 Statistical write voltage margin distributions of FinFET SRAM cells. $T = 90°C$

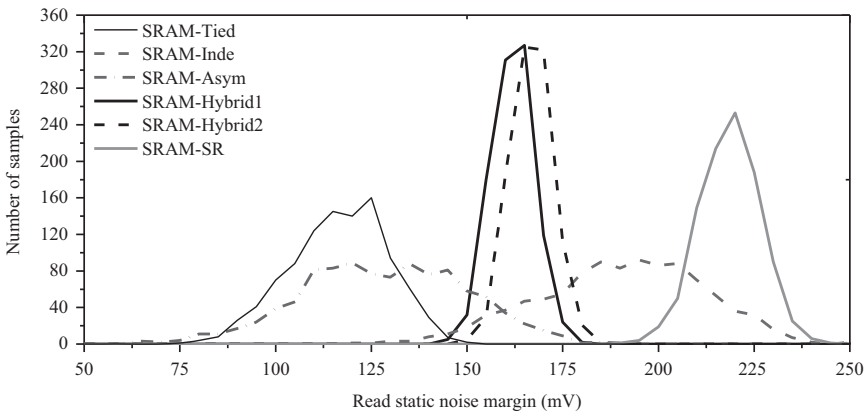


Figure 4.30 Statistical RSNM distributions of FinFET SRAM cells. $T = 90°C$

Table 4.1 Read access delay under process parameter fluctuations

SRAM topology	Read access delay (ps)		
	Mean	**Standard deviation**	**Worst-case**
SRAM-Tied	49.7	1.89	56.2
SRAM-Inde	113.8	8.19	128.0
SRAM-Asym	72.6	3.03	79.9
SRAM-Hybrid1	68.6	2.09	76.3
SRAM-Hybrid2	69.1	2.06	76.0
SRAM-SR	49.3	1.98	55.7

The gate control of the bitline access transistors during read operations is the weakest in SRAM-Inde among all the SRAM cells that are investigated in this chapter. SRAM-Inde is therefore the most sensitive to process variations during read operations. The standard deviation of read access delays with SRAM-Inde is 4.3×, 4.1×, 4×, 3.9×, and 2.7× higher as compared to SRAM-Tied, SRAM-SR, SRAM-Hybrid2, SRAM-Hybrid1, and SRAM-Asym, respectively, as listed in Table 4.1. Due to weaker bitline access transistors, the mean of read access delays with SRAM-Inde is increased by 2.3×, 2.28× 65.9%, 64.7%, and 56.7% as compared to SRAM-SR, SRAM-Tied, SRAM-Hybrid1, SRAM-Hybrid2, and SRAM-Asym, respectively.

Transistors with lower source/drain doping concentration are more sensitive to process parameter variations as compared to transistors with higher diffusion doping concentration when the region of operation is linear [31]. For suppressing the influence of process parameter variations on read delay fluctuations, the diffusion regions of pull-down transistor that stores a "0" in cross-coupled inverters are required to be heavily doped. A large portion of the right diffusion region of each pull-down transistor in cross-coupled inverters of the purely asymmetrical SRAM-Asym is undoped. The pull-down transistors in cross-coupled inverters of SRAM-Asym are therefore more sensitive to process parameter fluctuations. The standard deviation of read access delays with SRAM-Asym is 60.3%, 53%, 47.1%, and 45% higher as compared to SRAM-Tied, SRAM-SR, SRAM-Hybrid2, and SRAM-Hybrid1, respectively.

The gate underlaps of the transistors in the left-side inverter in SRAM-SR are longer as compared to SRAM-Tied, SRAM-Inde, SRAM-Hybrid1, and SRAM-Hybrid2. The data storage node on the left-side is however isolated from the single read bitline. Due to single-ended read operation, data disturbance is reduced with SRAM-SR as compared to other SRAM cells. The standard deviation of read speed distribution with SRAM-SR is therefore similar (difference less than 5.3%) to SRAM-Tied, SRAM-Hybrid1, and SRAM-Hybrid2.

The leakage current characteristics of asymmetrically gate overlapped/underlapped FinFETs (FinFET-Asym1) are more sensitive to process parameter variations as compared to FinFET-Sym and FinFET-Asym2. SRAM-Asym and SRAM-Hybrid1 therefore exhibit wider distributions of leakage power consumption as shown in Figure 4.28. Due to the weaker transistors in the left-side inverter,

Table 4.2 Leakage power consumption under process parameter fluctuations

SRAM topology	Leakage power consumption (nW)		
	Mean	Standard deviation	Worst-case
SRAM-Tied	53.9	9.20	94.4
SRAM-Inde	54.4	9.63	94.8
SRAM-Asym	142.0	38.68	315.8
SRAM-Hybrid1	76.1	21.13	197.4
SRAM-Hybrid2	49.1	8.34	92.3
SRAM-SR	36.9	6.57	64.4

Table 4.3 Write voltage margin under process parameter fluctuations

SRAM topology	Write voltage margin (mV)		
	Mean	Standard deviation	Worst-case
SRAM-Tied	226.3	3.54	214.3
SRAM-Inde	225.1	3.52	213.1
SRAM-Asym	236.2	5.87	217.6
SRAM-Hybrid1	228.2	4.13	208.2
SRAM-Hybrid2	217.5	4.36	201.3
SRAM-SR	249.5	5.00	230.9

SRAM-SR consumes the lowest average leakage power and exhibits the narrowest distribution of leakage power consumption. The standard deviation of leakage power consumption with SRAM-SR is 83%, 68.9%, 31.8%, 28.6%, and 21.2% lower as compared to SRAM-Asym, SRAM-Hybrid1, SRAM-Inde, SRAM-Tied, and SRAM-Hybrid2, respectively, as listed in Table 4.2. The mean of leakage power consumptions of SRAM-Asym, SRAM-Hybrid1, SRAM-Inde, SRAM-Tied, and SRAM-Hybrid2 are 3.8×, 2.1×, 47.4%, 46.1% and 33% higher, respectively, as compared to SRAM-SR. The worst-case leakage power consumption of SRAM-Asym, SRAM-Hybrid1, SRAM-Inde, SRAM-Tied, and SRAM-Hybrid2 are 4.9×, 3.1×, 47.2%, 46.6% and 43.3% higher, respectively, as compared to SRAM-SR, as listed in Table 4.2.

The transistors in the left-side inverter in SRAM-SR are weaker as compared to other SRAM cells. SRAM-SR thereby facilitates the transfer of new data into the cell. SRAM-SR offers the widest average write voltage margin under process parameter fluctuations. The mean of write voltage margin of SRAM-SR is 14.7%, 10.9%, 9.3%, 10.2%, and 5.6% wider as compared to SRAM-Hybrid2, SRAM-Inde, SRAM-Tied, SRAM-Hybrid1, and SRAM-Asym, respectively, as listed in Table 4.3. Due to the longer underlaps of transistors in the left-side inverter, the standard deviation of write

Table 4.4 Read static noise margin under process parameter fluctuations

SRAM topology	Read static noise margin (mV)		
	Mean (μ)	**Standard deviation (σ)**	**μ-6σ**
SRAM-Tied	116.5	12.67	40.5
SRAM-Inde	190.2	21.39	61.8
SRAM-Asym	128.4	21.69	−1.7
SRAM-Hybrid1	161.9	5.49	128.9
SRAM-Hybrid2	166.7	5.31	134.9
SRAM-SR	218.6	7.80	171.8

voltage margins in SRAM-SR is merely increased by 1.5 mV, 0.9 mV, and 0.6 mV as compared to SRAM-Tied/SRAM-Inde, SRAM-Hybrid1, and SRAM-Hybrid2, respectively. With the longest underlap in pull-up transistors, purely asymmetrical SRAM-Asym suffers from the most significant fluctuations of write voltage margin, as shown in Figure 4.29. The standard deviation of write voltage margin with SRAM-SR is 14.8% smaller as compared to SRAM-Asym.

The distributions of read static voltage margins of SRAM cells are shown in Figure 4.30. The eyes in the VTC of SRAM-SR are widest among all the SRAM cells that are investigated in this study. SRAM-SR is therefore more tolerant to process parameter fluctuations. The mean of RSNM of SRAM-SR is 87.6%, 70.2%, 35%, 31.1%, and 14.9% wider as compared to SRAM-Tied, SRAM-Asym, SRAM-Hybrid1, SRAM-Hybrid2, and SRAM-Inde, respectively, as listed in Table 4.4. SRAM-Inde also provides significantly larger average read data stability as compared to SRAM-Tied, SRAM-Asym, SRAM-Hybrid1, and SRAM-Hybrid2. Due to weaker gate control of bitline access transistors during read operations, however the distribution of read data stability with SRAM-Inde is significantly wider as compared to SRAM-Tied, SRAM-Hybrid1, SRAM-Hybrid2, and SRAM-SR, as shown in Figure 4.30.

The bitline access transistors in the hybrid SRAM cells are weaker during read operations as compared to SRAM-Tied. Furthermore, the pull-down transistors in cross-coupled inverters of the hybrid SRAM cells are stronger as compared to SRAM-Asym. The mean of RSNMs of SRAM-Hybrid2 is therefore 43.1% and 29.8% wider as compared to SRAM-Tied and SRAM-Asym, respectively, as listed in Table 4.4. The means of RSNMs of SRAM-Hybrid1 and SRAM-Hybrid2 are similar.

Robustness of memory circuits against process parameter variations is determined by the "yield margin" (μ-6σ) test [32, 33]. For a robust memory circuit, μ-6σ should be at least 4% of the power supply voltage. The yield margins of SRAM cells are listed in Table 4.4. μ-6σ of SRAM-Tied, SRAM-Inde, SRAM-Asym, SRAM-Hybrid1, SRAM-Hybrid2, and SRAM-SR are 40.5 mV, 61.8 mV, −1.7 mV, 128.9 mV, 134.9 mV, and 171.8 mV, respectively. Due to smaller μ/σ, purely asymmetrical SRAM-Asym is not stable under process parameter fluctuations. SRAM-SR is the memory cell with the strongest data stability under process variations.

4.6 Conclusions

Read and write voltage margin enhancement techniques for FinFET SRAM circuits are presented in this chapter. Six different FinFET SRAM cells are described and characterized. The single-ended read SRAM cell (SRAM-SR) with gate-underlap engineered tied-gate transistors provides the strongest data stability and write ability characteristics in active mode and consumes the lowest leakage power in idle mode. Furthermore, the single-ended SRAM cell is more tolerant to process parameter fluctuations in a 15 nm FinFET technology. SRAM-SR is therefore identified as the most effective circuit technique for achieving robust, ultra-low power, and variation tolerant memory subsystems in modern microprocessors that employ FinFET technology.

References

[1] Tawfik S. A., Kursun V. "Robust FinFET memory circuits with p-type data access transistors for higher integration density and reduced leakage power." *Journal of Low Power Electronics*. December 2009; Vol. 5(4): pp. 497–508.

[2] Kawasaki I., *et al.* "Demonstration of highly scaled FinFET SRAM cells with high-k/metal gate and investigation of characteristic variability for the 32 nm node and beyond." *Proceedings of the IEEE International Electron Devices Meeting*; USA, December 2008, pp. 237–240.

[3] Liu Z., Tawfik S. A., Kursun V. "An independent-gate FinFET SRAM cell for high data stability and enhanced integration density." *Proceedings of the IEEE International Systems on Chip (SOC) Design Conference*; Korea, September 2007, pp. 63–66.

[4] Chen Y.-H., Chan W. M., Wu W.-C., *et al.* "A 16nm 128Mb SRAM in high-k metal-gate FinFET technology with write-assist circuitry for low-V_{MIN} applications." *IEEE Journal of Solid-State Circuits*. January 2015; Vol. 50(1): pp.1–8.

[5] Yang J., Harris H. R., Hussain M. M., *et al.* "Enhanced performance and SRAM stability in FinFET with reduced process steps for source/drain doping." *IEEE Symposium on VLSI Technology, Systems, and Applications*; Taiwan, April 2008, pp. 20–21.

[6] Fossum J. G., Chowdhury M. M., Trivedi V. P., King T. J., Choi Y. K. "Physical insights on design and modeling of nanoscale FinFETs." *Proceedings of the IEEE International Electron Devices Meeting*; USA, December 2003, pp. 679–682.

[7] Yang J., Zeitzoff P. M., Tseng H. "Highly manufacturable double-gate FinFET with gate-source/drain underlap." *IEEE Transactions on Electron Devices.* June 2007; Vol. 54(6): pp. 1464–1470.

[8] *Process integration, devices, and structures (PIDS-2010)*. The International Technology Roadmap for Semiconductors (www.itrs.net). 2010.

[9] *Atlas user manual*. *Devedit user manual*. *Clever user manual*. www.silvaco.com. 2014.

[10] Rostami M., Mohanram K., "Dual-independent-gate FinFETs for low power logic circuits." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. March 2011; Vol. 30(3): pp. 337–349.

[11] Rabie M., Bahgat A., Ramadan K., *et al*. "A comparative analysis between FinFET semi-dynamic Flip-Flop topologies under process variations." *Proceedings of the IEEE International Conference on Energy Aware Computing*; Egypt, December 2011, pp. 1–6.

[12] Goel A., Gupta S. K., and Roy K., "Asymmetric drain spacer extension (ADSE) FinFETs for low-power and robust SRAMs." *IEEE Transactions on Electron Devices*. January 2011; Vol. 58(1): pp. 296–308.

[13] Salahuddin S. M., Jiao H., Kursun V. "Low-leakage hybrid FinFET SRAM cell with asymmetrical gate overlap/underlap bitline access transistors for enhanced read data stability." *Proceedings of the IEEE International Symposium on Circuits and Systems*; China, May 2013, pp. 2331–2334.

[14] Zhu H., Kursun V. "A comprehensive comparison of data stability enhancement techniques with novel nanoscale SRAM cells under parameter fluctuations." *IEEE Transactions on Circuits and Systems I*. April 2014; Vol. 61(4): pp. 2013–2021.

[15] Salahuddin S. M., Jiao H., Kursun V. "A novel 6T SRAM cell with asymmetrically gate underlap engineered FinFETs for enhanced read data stability and write ability." *Proceedings of the IEEE International Symposium on Quality Electronic Design*; USA, March 2013, pp. 353–358.

[16] Kranti A., Armstrong G. A. "Source/drain extension region engineering in nanoscale double gate SOI MOSFETs: novel design methodology for low-voltage analog applications." *Journal of Microelectronic Engineering*. December 2007; Vol. 84(12): pp. 2775–2784.

[17] Kadosh D., Gradner M. I., Duane M., *et al*. "Asymmetrical transistor structure," US Patent 6104064, 2000.

[18] Horiuchi T., Homma T., Murao Y., Okumura K. "An asymmetrical sidewall process for high performance LDD MOSFET's." *IEEE Transactions on Electron Devices*. February 1994; Vol. 41(2): pp. 186–190.

[19] Boeuf F., Skotnicki T., Monfray S., *et al.* "16 nm planar NMOSFET manufacturable within state-of-the-art CMOS process thanks to specific design and optimization." *Proceedings of the IEEE International Electron Devices Meeting*; USA, December 2001. pp. 637–640.

[20] Yu B., Chang L., Ahmed S., *et al.* "FinFET scaling to 10 nm gate length." *Proceedings of the IEEE International Electron Devices Meeting*; USA, December 2002, pp. 251–254.

[21] Miura N., Domae Y., Sakata T., *et al*. "Undoped thin film FD-SOI CMOS with source/drain-to-gate non-overlapped structure for ultra low leak applications." *Proceedings of the IEEE International SOI Conference*; USA, October 2005, pp. 176–177.

[22] Salahuddin S. M., Kursun V. "High-speed and low-leakage FinFET SRAM cell with enhanced read and write voltage margins." *Proceedings of the IEEE International Symposium of Integrated Circuits*; Singapore, December 2014.

[23] Choi Y. -K., King T. -J., Hu C. "Spacer patterning technology for nanoscale CMOS." *IEEE Transactions on Electron Devices*. March 2002; Vol. 49(3): pp. 436–441.

[24] Auth C., Allen C., Blattner A., *et al.* "A 22 nm high performance and low-power CMOS technology featuring fully-depleted tri-gate transistors, self-aligned contacts and high density MIM capacitors." *Proceedings of the IEEE Symposium on VLSI Technology*; USA, June 2012, pp. 131–132.

[25] Black B. Annavaram M., Brekelbaum N., *et al.* "Die stacking (3D) microarchitecture" *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*; USA, December 2006, pp. 469–479.

[26] Tawfik S. A., Kursun V. "Work-function engineering for reduced power and higher integration density: an alternative to sizing for stability in FinFET memory circuits." *Proceedings of the IEEE International Symposium on Circuits and Systems*; USA, May 2008, pp. 788–791.

[27] Tawfik S. A., Kursun V. "Portfolio of FinFET memories: innovative techniques for an emerging technology." *Proceedings of the IEEE International Systems on Chip (SOC) Design Conference*; Korea, November 2008, pp. 101–104.

[28] Liu Z., Tawfik S. A., Kursun V. "Statistical data stability and leakage evaluation of FinFET SRAM cells with dynamic threshold voltage tuning under process parameter fluctuations." *Proceedings of the IEEE/ACM International Symposium on Quality Electronic Design*; USA, March 2008, pp. 305–310.

[29] Tawfik S. A., Kursun V. "Stability enhancement techniques for nanoscale SRAM circuits: a comparison." *Proceedings of the IEEE International Systems on Chip (SOC) Design Conference*; Korea, November 2008, pp. 111–116.

[30] Gierczynski N., Borot B., Planes N., Brut H. "A new combined methodology for write-margin extraction of advanced SRAM" *Proceedings of the IEEE International Conference on Microelectronic Test Structures*; Japan, March 2007, pp. 97–100.

[31] Wang Y., Huang P., Xin Z., *et al*. "Impact of random discrete dopant in extension induced fluctuation in gate–source/drain underlap FinFET." *Japanese Journal of Applied Physics.* February 2014; Vol. 53(04EC05): pp. 1–4.

[32] Yu S., Zhao Y., Du G., *et al.* "The impact of line edge roughness on the stability of a FinFET SRAM." *Semiconductor Science and Technology*. February 2009; Vol. 24(2): pp. 1–9.

[33] Stolk P. A., *et al.* "CMOS device optimization for mixed-signal technologies." *Proceedings of the IEEE International Electron Devices Meeting*; USA, December 2001, pp. 215–218.

Chapter 5

Low-leakage techniques for nanoscale CMOS circuits

Manisha Pattanaik[1] *and Vijay Kumar Sharma*[1]

The previous few chapters focused on the variability issues of the nanoscale integrated circuits (ICs) for diverse applications including analog, radio frequency (RF), digital and memory ICs. The designs of such ICs are based on nanoscale bulk MOSFET and FinFET devices. The current chapter presents the leakage power dissipation which is an important issue of CMOS device in ultra-DSM regime. Leakage power dissipation is a major challenge especially for the energy efficient design of all battery-operated and portable real-time embedded systems in modern era. This chapter describes the various sources of leakage power, leakage and variability issues and provides a possible solution using some important leakage power reduction techniques at circuit/logic level for the state-of-the-art CMOS ICs.

5.1 Introduction

In the development of metal-oxide-semiconductor (MOS) ultra large scale integration (ULSI) technology, the basic elements of ULSI has been scaled down to deep sub-micrometer to nanoscale regimes. Over the last three decades, the very large scale integration (VLSI) designers have emphasized on miniaturization of the devices to get better speed and area of the logic circuits. The chip area and speed of the circuits are improved by scaling the sizes of a semiconductor device. The reduction in dimensions of semiconductor device gives better results in terms of device density on same chip, thereby enhancing the functionality of the integrated circuits (ICs). The rising device density on a chip has increased the power consumption. Power density has kept growing with each new technology node scaling. More power density raises the problem of heating. Extra cooling systems are needed to reduce the heat problem. These cooling systems may degrade the reliability and functionality of the logic circuits. The power consumption of a logic circuit is proportionally related to apply power supply voltage. Hence, other solution of heat problem is applying low power supply voltage. Power supply voltage

[1]ABV – Indian Institute of Information Technology and Management, Gwalior, India

and threshold voltage of a device are the critical parameters. Both parameters should be reduced for performance improvement. The reduction of threshold voltage of a device adds the issue of leakage current. Sub-threshold leakage current which is the dominant leakage current component increases exponentially with threshold voltage scaling.

In current years, the evaluation of portable systems and demand of nanoscale complementary metal-oxide-semiconductor (CMOS) VLSI fabrication technologies lead to high power dissipation which is an issue of concern for many VLSI systems. The broad necessity of battery operated portable applications need to explore the low power VLSI research field. Low power dissipation design increases the reliability and reduces the cooling cost of the portable systems. In deep sub-micron regime, the static or leakage power dissipation component is the dominant part of power dissipation. Leakage power dissipation affects the circuit performance differently depending on external conditions, operating modes and logic families. Responding to this challenge, several leakage power reduction techniques at different abstraction levels are proposed to reduce static power for portable systems and make them more energy efficient. Process variability is considerably increasing with each new technology node scaling and causes performance fluctuations. Parameter variations are affecting the leakage current in several ways in ultra-DSM regime.

This chapter is devoted to the leakage reduction techniques for nanoscale CMOS circuits. In this chapter, initially the impact of device scaling is presented. In addition, leakage power dissipation, leakage current components, leakage challenges, variability issues and aware designs are briefly described. Finally, the various transistor/logic level leakage power reduction techniques and its leakage analysis are presented.

5.2 Device scaling

Transistor device dimensions continuously scales down since last three decades, the number of transistors on chip has thus increased to integrate more applications in small area and to improve the performance of circuits. This increases the device density with reducing propagation delay on single silicon buffer.

G. E. Moore who is one of the co-founders of Intel Corporation has predicted that the number of transistors that can be integrated on a single chip doubles approximately after every two years without increasing the cost and has become known as Moore's law [1]. Moore observed that the scaling of the transistor's dimensions with larger die sizes would lead to cheaper and higher functionality ICs. Figure 5.1 shows the number of CPU transistors has increased by 2X and feature size has decreased by 0.7X and follows the Moore's law [2]. Technology scaling not only increases the transistor density but also increases the switching speed of logic circuits.

Robert Dennard at IBM has published a theory for Metal Oxide Semiconductor Field Effect Transistor (MOSFET) scaling. He considered three primary variables of a transistor scaling; dimensions, supply voltage and doping to invent a new smaller

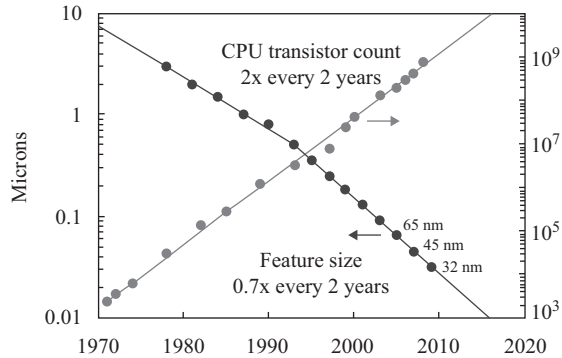


Figure 5.1 CPU transistor counts and transistor feature size 1970–2020 (Source: K. J. Kuhn, 2009, p.1 [2])

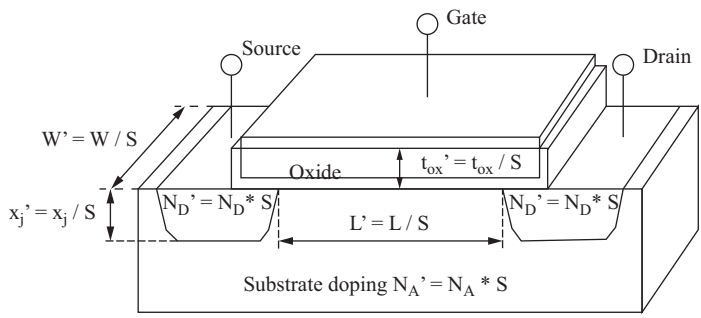


Figure 5.2 Scaling of a typical MOSFET by a factor of S

transistor [3, 4]. The growth of portable electronic devices demand low power space efficient circuit design. Moore's law and Dennard's scaling theory shaped the semiconductor industry to continue reduction of the minimum dimension for a silicon wafer [5]. The length of the gate of a MOS transistor refers the smallest feature size for the logic circuits. Feature size is used to characterize a technology node. All the dimensions of MOSFET are reduced by a scaling factor of 'S'. Minimum dimension is the feature size of the transistors. Figure 5.2 illustrates the key dimensions of a typical MOSFET.

Moore's law does not provide any mathematical and physical theory of scaling of devices. Roughly, at each new technology node, the line widths of circuit layouts are 70% of the previous node in order to reduce the circuit size by 2 (70% of previous line width means $\approx$ 50% reduction in area, i.e., $0.7 \times 0.7 = 0.49$). The speed of ICs has been increased nearly 30% at each new technology node [6–8]. Since nearly twice as many circuits can be fabricated on each wafer with each new technology node, the cost per circuit is reduced significantly. Other MOSFET parameters such as gate oxide thickness and the power supply voltage are also reduced with scaling. The smaller transistors and shorter interconnects lead to smaller capacitances. These

scaled device dimensions guide the circuit delays to drop. Dennard's scaling theory suggested 30% circuit delay improvement per new technology node if 'S' = 1.4.

There are three types of scaling strategies: constant voltage scaling, constant field scaling and generalized scaling. The scaling of MOS transistor by a scaling factor is the systematic and proportional reduction of overall area while preserving the geometric ratios found in larger devices.

5.2.1 Constant voltage scaling

All the dimensions of MOSFET are reduced by a factor of 'S' as listed in Table 5.1 while keeping the power supply voltage and the terminal voltages remain unchanged from one technology node to next node. The doping densities are increased by a factor of 'S^2' in order to preserve the charge-field relation. The electric field is increased by 'S', thereby improving the carrier velocity. The power density is increased by 'S^2' that may cause serious reliability problems for the scaled devices. The carrier velocity saturates beyond a critical lateral electric field [9]. Therefore, scaling of the power supply voltage became essential since the 0.8 μm technology node [10]. The constant voltage scaling is usually preferred over constant field scaling when necessitate multiple power supply voltages.

5.2.2 Constant field scaling

In constant field scaling method, all the horizontal and vertical dimensions of a transistor as well as the power supply voltage are scaled down by a factor of 'S'. The doping densities are increased by the factor 'S' as listed in Table 5.1 in order to preserve the magnitude of the internal electric field. Constant field scaling allows more things to occur faster at the equal energy cost, and it is economically attractive

Table 5.1 Influence of scaling on MOS device characteristics

Parameter	**Constant field**	**Constant voltage**	**Generalized**
Channel length (L)	1/S	1/S	1/S
Channel width (W)	1/S	1/S	1/S
Gate oxide thickness (t_{OX})	1/S	1/S	1/S
Electric field (E)	1	S	U
Power supply voltage (V_{DD})	1/S	1	U/S
Threshold voltage (V_{TH})	1/S	1	U/S
Doping densities (N_A, N_D)	S	S^2	US
Oxide capacitance (C_{OX})	S	S	S
Drain current (I_D)	1/S	S	U/S
Delay (τ)	1/S	$1/S^2$	1/S
Power dissipation (P_{DISS})	$1/S^2$	S	U^2/S^2
Power density (P/Area)	1	S^3	U^2
Power delay product (PDP)	$1/S^3$	1/S	U^3/S^3

if the manufacturing cost per square area grows only modestly [11]. The saturation current is degraded with the constant field scaling method [12]. The scaling of power supply voltage affects the threshold voltage of MOS transistor. It is necessary to scale the threshold voltages according to the power supply voltage scaling in order to maintain the performance of the device. The threshold voltage is lowered for enhanced device current, thereby leading to the exponential increase of sub-threshold leakage current [13]. It is clear that constant field scaling reduces both the drain current and the supply voltage by a factor of 'S'. Therefore, the power dissipation of the transistor decreases by a factor of 'S^2'.

5.2.3 Generalized scaling

The primary limit of constant field scaling is the non-scaling of the sub-threshold slope and large gate leakage [14]. The threshold voltage cannot be scaled at the same pace as the other device parameters in deep sub-micrometer MOSFETs. Constant field scaling cannot be used for future scaled devices because of several fundamental limitations of scaling. A new scaling strategy called generalized scaling can be used for that purpose [15]. In generalized scaling method, a scaling factor 'U' ($1 < U < S$) is used for the MOSFETs to achieve the same speed as the constant field scaling method [16]. Generalized scaling increases the power density by 'U^2' which have an effect on chip packaging and systems design. Hence, there is practical limit for the utilization of generalized scaling.

5.3 Power dissipation

There are two types of power dissipation in CMOS digital circuits: dynamic power and static power dissipation. Switching power and short-circuit power dissipations are the two components of dynamic power dissipation. Total power dissipation of a logic circuit is given as [17]

$$P_{TOTAL} = P_{DYN}(P_{SWITCH} + P_{SC}) + P_{LEAK} \tag{5.1}$$

where P_{DYN} is dynamic power dissipated during the logic transitions and P_{LEAK} is power dissipated during steady-state period. P_{DYN} comprises two components: the switching power P_{SWITCH} and the short-circuit power P_{SC}. Leakage power dissipation affects the performance of logic circuits dominantly in nanoscaled technologies.

5.3.1 Leakage power dissipation

Leakage power dissipation is the off-state current flowing between power supply voltage and ground during steady state input levels. Generally, the dynamic power is the dominant component of total power dissipation and the static power part is negligible. But this will be totally wrong in the case as the CMOS technology scales down below 100 nm technology nodes [18]. Figure 5.3 illustrates the leakage power dissipation component when input voltage is at steady state logic '0' or '1' levels.

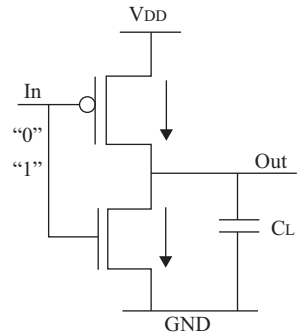


Figure 5.3 Leakage power dissipation in CMOS inverter circuit

Leakage power is one of the most crucial design components below 100 nm technologies which must be efficiently controlled in order to utilize the performance advantages of these technologies. Leakage power can be given as

$$P_{LEAK} = V_{DD} \cdot \sum_{i=1}^{N} I_{LEAKi} \tag{5.2}$$

where P_{LEAK} is leakage power or static power, I_{LEAKi} is off-state current or leakage current of ith transistor and N is total number of transistors in a circuit.

5.3.2 Leakage current components

Leakage current is the summation of number of leakage components present in a MOS device. Different components of leakage current in NMOS device are sub-threshold leakage I_{SUB}, gate leakage I_G, hot-carrier injection (HCI) leakage I_H, gate induced drain leakage (GIDL) $I_{GIDL,}$ junction leakage $I_{REV}$and punch-through leakage I_P as shown in Figure 5.4 [19].

Sub-threshold leakage (I_{SUB}) is the weak inversion current due to minority carriers diffusion between source and drain regions of the transistor. When gate voltage of a transistor is less than threshold voltage then that transistor is not completely turned-off and small current flows between source and drain terminals [20]. The amount of sub-threshold leakage current is very small for above 100 nm technology nodes. But it increases significantly when technology nodes scaled down with transistors dimensions [21]. It dominates the total off-state current. Sub-threshold leakage current must be managed and minimized for energy efficient portable nanoscaled devices. The sub-threshold current is a function of the thermal equivalent voltage, supply voltage, device size and the process parameters.

Device scaling forces to reduce gate oxide layer thickness between the gate and channel to increase the channel conductivity and performance. Gate oxide thickness reduction leads to lowering of gate barrier potential and some positive charges get stuck there for some positive gate voltage. Therefore, oxide layer thickness reduction

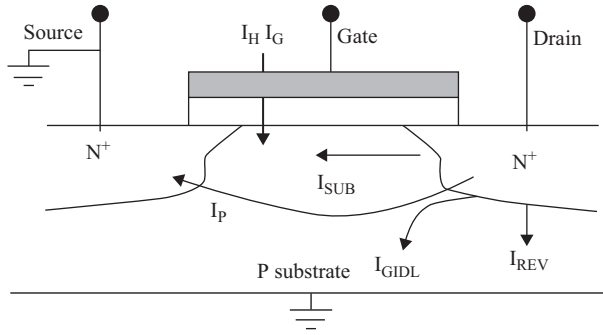


Figure 5.4 Leakage current components in NMOS transistor (Source: N. Ekekwe, 2006, p.853 [19])

results gate tunnelling current (I_G) from gate to substrate and from substrate to gate in very deep sub-micron technologies [22].

The charge carriers can get sufficient energy from the high lateral electric field near the Si/SiO_2 interface in a short-channel transistor to cross the interface potential barrier. At high lateral fields within the channel, electrons involve in impact ionization in the channel. It induces leakage current I_H [23]. The threshold voltage is changed due to occupied oxide traps. The amounts of electrons are large to enter the oxide layer because of lower barrier height as compared to holes. I_H depends on apply power supply voltage as the higher value of V_{DD} voltage heats the carriers. The scaling of V_{DD} reduces this problem.

If the gate voltage of a MOS transistor is like that it creates accumulation layer underneath SiO_2 layer then channel region has almost same potential as the substrate. The channel region behaves highly doped than substrate and decreases the width of depletion layer at the surface. The further application of drain voltage narrowed the depletion layer at drain diffusion region under the gate. GIDL current (I_{GIDL}) flows by the minority carriers underneath the gate due to high electric field of a MOS transistor [24].

The reverse-bias p-n junction formation between the source substrate and drain substrate regions brings junction leakage I_{REV} current. The area of the source/drain diffusion regions, the high doping concentration, minority carrier diffusion/drift near the edge of depletion region, mobile carrier generation in depletion region of the pn reverse-bias junction affects I_{REV} [25].

The separation between the depletion layers at drain substrate and source substrate junctions is decreased in short-channel devices while keeping doping constant. The depletion regions of drain/source substrate junctions are pushed to merge each other as channel length is reduced. Punch-through leakage current (I_P) occurs when both depletion layers are merged [23]. I_P current chiefly depends on the applied drain voltage and on the source/drain junction depths. The additional implants are used to control the I_P current.

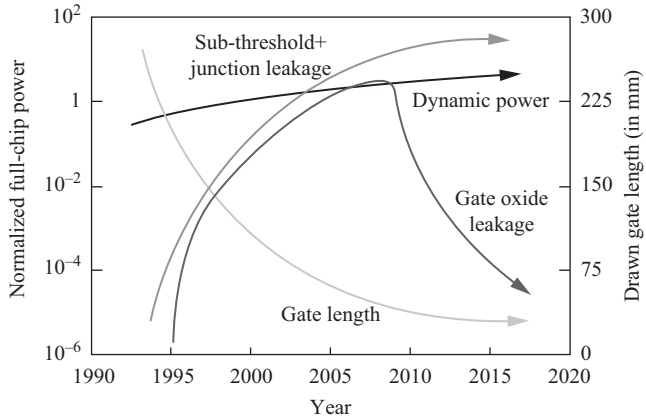


Figure 5.5 Power dissipation and physical gate length trends (Source: Z. Abbas, 2014, p.180 [28])

5.4 Issue of leakage current

The power supply voltage and threshold voltage scaling of a MOS transistor are the critical parameters in order to maintain the performance and switching speed of MOS device. The supply voltage must be scale down as new technology generation in order to keep power dissipation, power delivery costs under control. The threshold voltage of the transistor must be reducing proportionally with reducing the supply voltage of the device so that it can maintain the performance of the device [26]. The leakage current increases dramatically with each new technology generation. Some researchers predict a 7.5X increase in the leakage current and a 5X increase in total energy dissipation for every new microprocessor chip generation [27]. As the leakage current increases more rapidly with technology scaling, it will become more and more effective part of the total power dissipation. Figure 5.5 indicates the importance of leakage power with reducing the physical gate length [28].

5.5 Variability issues and aware design

The rapid scaling of silicon technology is leading to a considerable increase in process variability [29, 30]. A chip can stay in ideal state or in operating state several times during its life period. Working conditions of the chip varies from time to time like supply voltage, temperature, different set of input voltages, etc. It is not possible to meet all specifications of the chip in each time. Large leakage power in scaled deep submicron regime is heating the die and results hot-spot. The hot-spot degrades the reliability and is the primary source of thermal runaways of the package [31]. Parameter variations occur due to lack of control over the fabrication process.

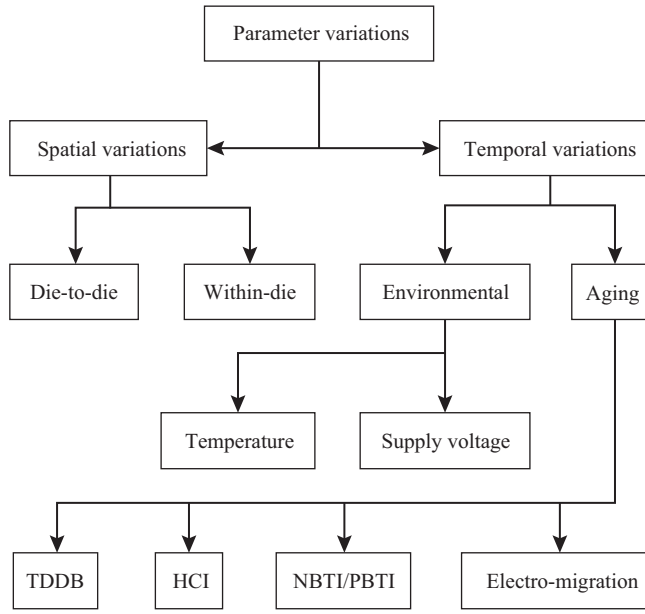


Figure 5.6 Sources of parameter variations in integrated circuits

It is impossible to sustain the exact level of manufacturing uniformity below 100 nm technology nodes. Hence, model characteristics of devices are changed. It becomes particularly important at smaller technology nodes and results in uncertainties in speed and power characteristics of ICs [32]. Process variations introduce the statistical inter-die/intra-die variations in physical properties of transistors which result in degrading the performance and logic functionality of ICs [33]. The classification of parameter variations is shown in Figure 5.6

Parameter variations have two broad categories: spatial and temporal variations. Spatial process variations are static variations and defined at $t = 0$ time while temporal variations are dynamic variations or $t \neq 0$ variations. Inter-die variations cause electrical parameter variations such as fluctuations in length, width, threshold voltage which come from different runs. Intra-die variations are the physical parameter variations like line edge roughness, random dopant fluctuations and are the within-die variations.

Spatial variations are the process variations [34, 35]. Imperfect control over the fabrication process leads process variations and varies from foundry to foundry. Process variations can be reduced by the better control of the process. Process variations result in fluctuation of the V_{TH} value. Inter-die variations are systematic variations while Intra-die variations are random variations. Systematic inter-die variations affect the adjacent transistors on a chip with identical shift from nominal value. These variations are deterministic and can be predicted in advance by analyzing the layouts. All devices on a chip can be easily modelled as having same process parameters. Random

intra-die variations affect the adjacent transistors on same chip with different shifts. These variations are random in nature and hence cannot be predicted in advance before manufacturing [36]. Inter-die variations can be compensated by using forward and reverse-body bias techniques. It is more difficult to control intra-die variations due to random nature of variations.

Environmental and aging related variations are temporal variations. The power supply voltage and temperature are the two environmental variations factors. Environmental conditions play a vital role in the process of designing the ICs. Each circuit block on a chip has different activity and creates different power densities that result in temperature gradient. The on-state and off-state currents are varied since ICs feel thermal behavior due to temperature gradient. Many device parameters are very sensitive to the temperature gradient. The sub-threshold leakage current varies significantly for a change in device parameters. The dimensions of a device are scaled drastically when moving from higher technology node to lower node as compared to apply supply voltage scaling. The supply voltage cannot be scaled significantly without degrading reliability and exponentially increasing leakage power due to threshold voltage. Aging variations refer to the change in device characteristics over time. Negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), HCI, time dependent dielectric breakdown (TDDB) and electro-migration are the aging related variations. The electrical characteristics of ICs tend to degrade after a long period of operation. Body biasing is the controlling parameter for mitigating the variability effect at circuit level design [37].

5.6 Leakage reduction techniques

It is required to use leakage reduction technique for designing low leakage circuits. Leakage reduction technique can be applied at different abstraction levels like, fabrication (technology or device) level, transistor/circuit/logic level, system level, algorithm (behavior) level and architecture (structure) level. This chapter presents the transistor/circuit/logic level leakage reduction techniques.

5.6.1 MTCMOS technique

Multi-Threshold CMOS (MTCMOS) is a valuable circuit-level methodology that provides high characteristics in the active mode and saves leakage power during the standby mode. This technique comes in the category of gated power supply leakage reduction. The basic principle behind the MTCMOS technique is to use low threshold transistors to design the logic gates where the performance is essential, while the high threshold transistors (sleep transistors) are used to effectively isolate the logic gates in standby state and limit the leakage dissipation [38]. Schematic arrangements of MTCMOS technique is given in Figure 5.7.

In the MTCMOS technique, logic cell is supplied by a virtual power rail. The low threshold logic gates are attached to a virtual ground line. The virtual ground line

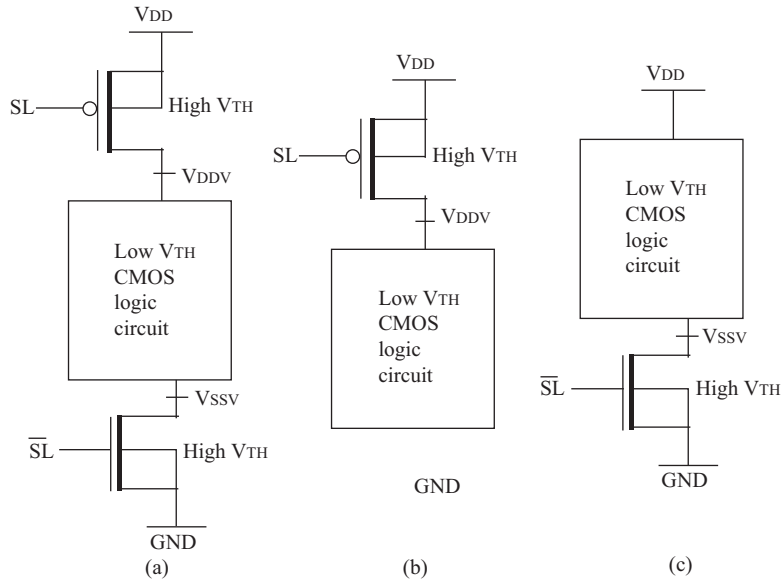


Figure 5.7 MTCMOS scheme (a) Insert NMOS & PMOS (b) insert PMOS (c) insert NMOS (Source: S. Mutoh, 1995, p.849 [38])

is connected to the actual ground through a high threshold sleep transistor. Similarly, the virtual power line is connected to the actual power supply through a high threshold sleep transistor [39]. In the active mode operation, the high threshold sleep transistors are turned on, while in the standby mode of operation, the high threshold sleep transistors are in cut-off, and hence reducing the leakage current.

The key points to the success of this MTCMOS scheme are that

a. In the active mode the sleep transistors are conduct (SL = '0') so having low on resistance cause maximum flow of current. In normal mode the sleep transistors have high threshold voltages. This attained the high performance of the logic circuit.
b. In sleep mode the sleep transistors are in cut-off condition (SL = '1') so the logic cells are connected to virtual power supply (V_{DDV}) and virtual ground (V_{SSV}). The threshold voltage of sleep transistors are chosen such that the leakage in the sleep mode of the MTCMOS logic cells is significantly smaller.
c. Choose the proper size of the sleep transistors so they could not increase the layout area and the power dissipation. This trade-off becomes even more apparent in the deep sub-micron regime. Therefore, proper sizing of sleep transistor is a key element to efficiently design complex MTCMOS circuits. MTCMOS technique provides poor performance at low power environment. Boosted Gate MOS scheme overcomes this problem [40].

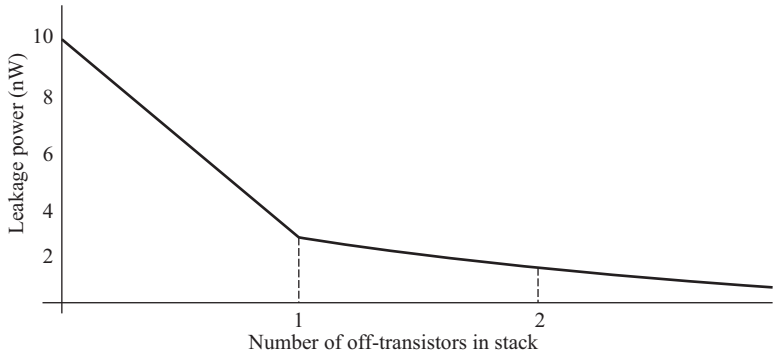


Figure 5.8 Leakage power vs number of off-transistors in stack

In MTCMOS technique, only one type (NMOS or PMOS) of sleep transistor is enough for leakage control. Figure 5.7(b) and (c) shows the PMOS insertion and NMOS insertion schemes, respectively. The NMOS insertion scheme is preferable, since the NMOS has smaller on-resistance at the same width; therefore, it can be sized smaller than corresponding PMOS. However, MTCMOS can only reduce the standby leakage power, and the large inserted MOSFETs can increase the area and delay. It reduces the noise margin or in worst case may result in complete failure of the gate. Here need of two different oxide thicknesses for two different threshold voltages causes a serious problem. The deposition of two different oxide thicknesses is a complicated task [41].

The MTCMOS circuits suffer from high energy overhead during the transitions between the active and standby modes. It is due to charge stored by the parasitic capacitances at the virtual rail line. A new circuit technique is used to lower the energy overhead of these mode transitions. The charge stored at the "virtual power" and "virtual ground" lines are recycled during the mode transitions [42].

5.6.2 Forced stack technique

This technique inserts extra series connected transistor in the pull-down or pull-up path of a gate and turns it off in standby mode. The extra transistor is turn-on during normal operation. This provides a substantial savings in leakage current during standby mode. There may be number of series connected transistors in stack but at least one transistor must be off [43]. Number of transistor stacks reduces the leakage current with high magnitude.

Figure 5.8 shows the leakage power vs number of off-transistors in a stack. There is a large difference in leakage power between one off-transistor and two off-transistors. Turning off three transistors does improve leakage power.

However, the extra stack transistor makes the drive current of the forced stack gates lower, resulting in increased delay. Hence, this technique is only usable for noncritical paths. Noncritical path shows the effective circuit management.

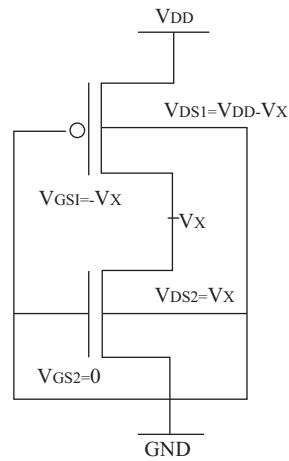


Figure 5.9 Two transistor stack

In noncritical path the transistors having higher threshold voltage thus reduce leakage current because leakage current inversely depends on threshold voltage. Stacking of transistors is also called forced NMOS or forced PMOS techniques [44].

In stacking technique the amount of leakage currents of nanometer CMOS circuit varies depending on the input signal pattern of the transistor stacks. So this is also called as input pattern control technique and corresponding input signal pattern is called minimum leakage vector. Here we choose the best combinations of input signal pattern, which sets up the minimum leakage current during standby mode [45]. By changing the internal logic gates circuit reduction in leakage may be achieved. Several techniques have been proposed to generate the minimum leakage test pattern. One simple straightforward method to find a best low leakage input pattern is to count all combinations of primary inputs. This is not a good method because n input signals have 2^n combinations. This cause circuit complexity and required more computational time. For finding the best input pattern we can use random search–based genetic algorithm.

For understanding the concept of transistor stack we consider a two transistor stack shown in Figure 5.9. Here we find out the gate to source voltage, drain to source voltage and substrate to source voltage of transistor connected in series. $V_X$is the intermediate node voltage. V_X is positive due to small drain current. The leakage current is dependent on the voltages of all the four terminals of the transistor. Transistor stacking technique exploits the dependence of leakage current on the source terminal voltage V_S. If we increase the V_S of the transistor, the leakage current reduces exponentially.

Positive potential at intermediate node (V_X) has three effects:

1. Due to positive source voltage V_X, gate to source voltage V_{GS1} becomes negative hence main component of leakage (sub-threshold current) reduces significantly.

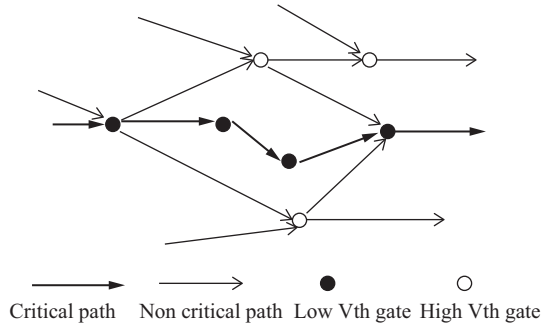


Figure 5.10 Critical and non critical paths (Source: K. Roy, 2003, p.320 [23])

2. Due to positive V_X, body to source voltage V_{BS1} becomes negative (body effect), resulting in increase of threshold voltage of upper transistor and hence reducing the leakage current.
3. Due to positive V_X, drain to source voltage V_{DS1} decreases, resulting increase in threshold voltage of upper transistor and hence reducing the leakage current.

5.6.3 Dual threshold CMOS (DTCMOS) technique

MTCMOS circuits require the insertion of extra series connected high threshold devices which limit the leakage currents during the standby mode. However, these sleep transistors are difficult to size correctly, and being in series with the pull-down and pull-up path will always degrade performance. Another circuit level design style is dual-threshold voltage [46].

In logic circuits, high threshold voltage can be assigned to some transistors in the non critical paths so as to reduce the leakage current, while the performance is maintained by using low threshold voltage transistors in the critical path. No additional circuitry is required, and both high performance and low leakage can be achieved simultaneously. Figure 5.10 illustrates the basic idea of a DTCMOS circuit.

Dual threshold voltage CMOS has the same critical delay as the single low threshold voltage MOS circuit, but the transistors in non-critical paths can be assigned high threshold voltage to reduce leakage power. DTCMOS is effective in reducing leakage power during both standby and active modes. DTCMOS based flip-flop can reduce leakage power dissipation with reducing in dynamic power dissipation and silicon area [47]. There are many design techniques have been proposed, which consider the sizing of high threshold voltage transistor in dual threshold voltage design to improve performance, and to reduce leakage power [48].

5.6.4 SCCMOS (super cut-off CMOS) technique

This is the multi-voltage (MVCMOS) technique. In MTCMOS technique high threshold sleep transistors are used for reducing the leakage current. While in this technique

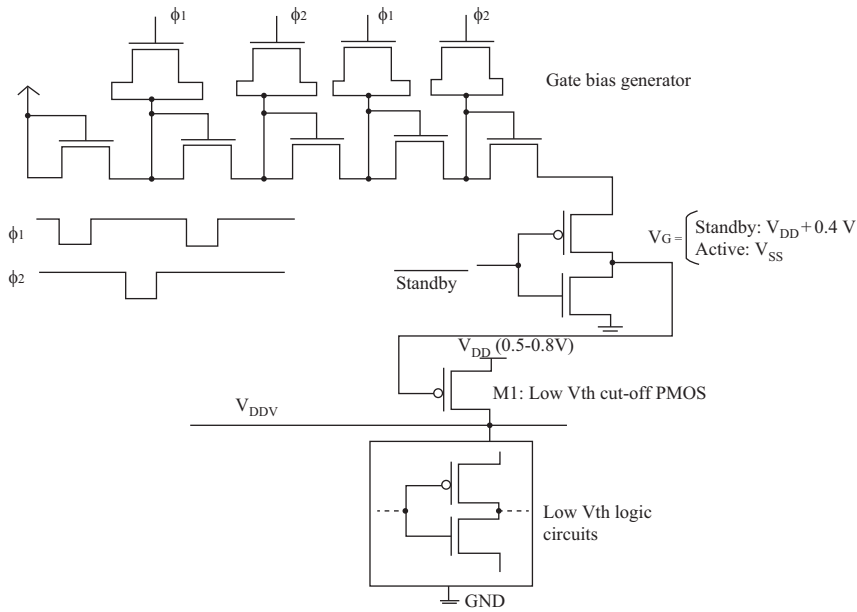


Figure 5.11 SCCMOS technique (Source: H. Kawaguchi, 2000, p.1498 [50])

low threshold same as logic block's threshold voltage sleep transistors are used [49]. SCCMOS used to solve the problem of the high threshold transistor in MTCMOS technique. This is the gated power supply leakage reduction technique. As we know the dynamic power dissipation is proportional to the square of supply voltage. So by applying this method the both leakage power and dynamic power dissipation components are reduced fast. The basic idea behind this method is to turn-off the transistor fully which connect the main circuit to power supply or ground in standby mode. This is achieved by connecting the gate of the PMOS sleep transistor to a voltage which is higher than power supply V_{DD}. This leads to a more positive gate to source voltage for a PMOS transistor so PMOS sleep transistor always in cut-off state. The gate of the NMOS sleep transistor is connecting to a voltage which is lower than ground level. This leads to a more negative gate to source voltage for a NMOS transistor. By doing this both sleep transistors are in super cut-off state [50]. Therefore this method is called super cut-off CMOS technique and hence, the sub-threshold current reduces exponentially with increasing this (these) voltage(s). The increase or decrease in the gate voltage is obtained using an extra circuitry called charge pump circuit which is shown in Figure 5.11. The overall circuitry for gate bias generator is called charge pump circuit.

In Figure 5.11, the low threshold voltage cut-off PMOS, M1, whose threshold voltage is 0.1–0.2 V, is inserted in series to the logic circuits consisting of low threshold voltage MOSFETs. The low threshold voltage provides the lower propagation delay hence improve the speed of the logic circuits. In active mode of operation the gate

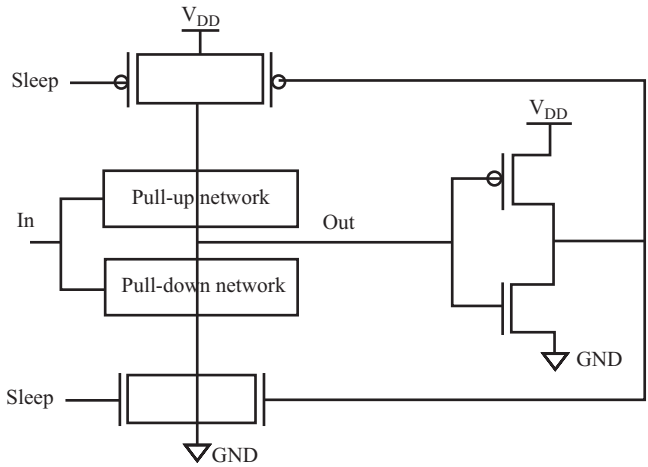


Figure 5.12 Leakage feedback circuit (Source: S. H. Kim, 2006, p.368 [54])

voltage of M1, is grounded to turn-on the M1 transistor. While in standby mode of operation V_G is overdriven to $V_{DD}+ 0.4\,V$ to completely cut-off the PMOS transistor M1. This is because the low threshold voltage of 0.1–0.2 V is lower by 0.4 V than conventional high threshold voltage (0.5–0.6 V), and thus this overdriven mechanism can maintain the standby current level. A MOS charge pump circuit is used to generate this overdrive voltage of $V_{DD}+ 0.4\,V$.

The charge pump circuit pumps charges induced by an oscillator circuit to or from the bulk capacitance, C_b [51]. Here a strong charge pump circuit is required to sufficiently increase the gate voltage in an acceptable time. If threshold voltage is lower than 0.1–0.2 V or negative, V_G should also lower as long as there is no problem of gate oxide reliability or GIDL. This charge pump circuit may cause extra power consumption, area requirement and delay. These are the main drawbacks of this method [52].

5.6.5 Leakage feedback technique

The leakage feedback technique uses two additional low threshold transistors to preserve the logic state in idle or standby mode without using the high threshold voltage devices. These two additional transistors are driven by the output of an inverter which is driven by output of the logic circuit [53]. As shown in Figure 5.12, a low threshold PMOS transistor is placed in parallel to the sleep high threshold PMOS transistor and a low threshold NMOS transistor is placed in parallel to the sleep high threshold NMOS transistor. These two transistors are driven by the output of the inverter which is driven by the output of the logic circuit. During active mode both sleep transistors and both parallel connected low threshold transistors (helper transistors) are turn on. During sleep mode, sleep transistors are turn off and one of the transistors in parallel

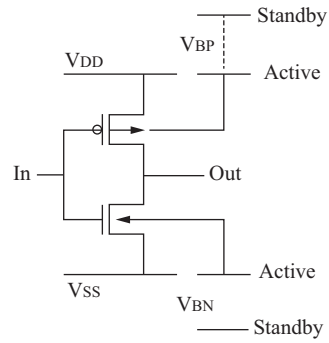


Figure 5.13 VTCMOS schematic (Source: K. Roy, 2003, p.321 [23])

to the sleep transistors, i.e., helper transistor keep the connection with the appropriate power rail and therefore reduce the leakage current.

Helper transistors are used to hold strong logic low and strong logic high during standby mode. By using two helper transistors the output of logic block will always be driven to one power rail or another, and will not be float longer in standby mode.

5.6.6 Variable threshold CMOS (VTCMOS) technique

Variable Threshold CMOS (VTCMOS) is a circuit design technique that has been developed to reduce standby leakage currents by using a triple well process technique where the device threshold voltage is dynamically adjusted by biasing the body terminal. VTCMOS is the extension of DTMOS technique. VTCMOS circuit essentially uses low threshold voltage transistors, and the substrate bias voltages of the transistors that are generated by the variable substrate bias control circuit. When the VTCMOS inverter circuit in Figure 5.13 is operating in its active mode, the VTCMOS inverter transistors work as conventional CMOS transistors and do not have any body bias effect. When the logic circuit is in the standby mode, the substrate bias control circuit generates a lower substrate bias voltage for the NMOS transistor and a higher substrate bias voltage for the PMOS transistor. This body biasing effect increases the magnitude of the threshold voltage of the transistors in the standby mode. The threshold voltage of the device depends upon the body to source voltage. Threshold voltage varies according to the relation given in (5.3) [55].

$$\Delta \mathrm{Vth} = \gamma \mathrm{V_{BS}} \tag{5.3}$$

where γ is the body bias coefficient.

Therefore, the leakage power dissipation in the standby mode can be considerably reduced with this circuit design technique. But, with continuously technology scaling, it has been showed that the effectiveness of VTCMOS reduces as the channel lengths become smaller, or the threshold voltage values are lowered (Vth roll-off effect). Also,

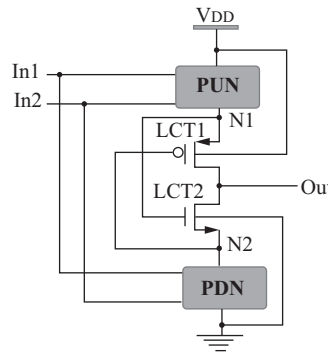


Figure 5.14 LECTOR technique (Source: N. Hanchate, 2004, p.198 [57])

VTCMOS is basically more challenging for reliability since the high voltage across the oxide decreases the lifetime of the device.

In addition, this method could be applied in active mode operation to optimize circuit performance by dynamically tuning the threshold voltage based on workload needs. By proper body bias tuning, the circuit is able to operate at the minimal active leakage power [56].

5.6.7 LECTOR technique

In LECTOR (LEakage Control TransistOR) technique insert two leakage control transistors (LCTs) (a PMOS and a NMOS) in each CMOS gate as shown in Figure 5.14. In this technique each LCT is controlled by the source of the other LCT [57]. Since one of the LCTs is always near its cut-off for any combination of input signals. It causes decrease in current in the path from V_{DD} to ground. LECTOR is single threshold; vector independent method which requires only two transistors for every path in a circuit. The concept of this technique came behind the stacking of the transistors in the path of power supply to ground. When one or more transistors are in cut-off mode then they behave as large value resistance and cause negligible leakage current. The important feature of LECTOR is that it works effectively in both active and standby mode of the circuit, resulting in better leakage reduction technique.

5.6.8 Sleepy stack technique

This technique mixes the sleepy approach and stack approach. For stacking approach the one top most transistor of pull-up network and one top below transistor of pull-down network divides into two equal half size transistors. For sleepy approach one PMOS transistor is connected to the parallel of one of divided transistor in pull-up network and one NMOS transistor is connected to the parallel of one of divided transistor in pull-down network [58]. The schematic view of this technique is shown in Figure 5.15.

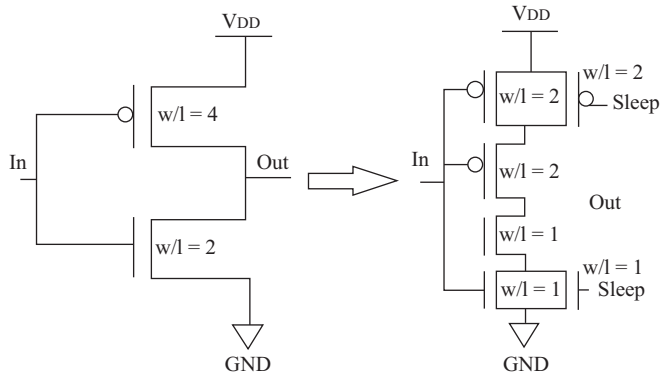


Figure 5.15 Sleepy stack CMOS inverter (Source: J. C. Park, 2006, p.1252 [58])

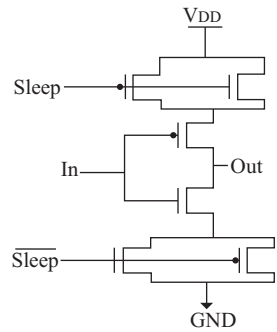


Figure 5.16 Dual sleepy stack (Source: M. S. Islam, 2010, p.90 [59])

This technique reduces leakage current in standby mode by managing the power supply of the main logic block. In sleep mode, sleep transistors are turn-off and stacked transistors suppress leakage current. Due to parallel connection of sleep transistors, they reduce the overall path resistance in active mode. This is also beneficial for significantly delay declined in active mode. Area is the focused concern of this technique since one transistor replaced by three transistors. Additional circuitry required for sleep signals. Another technique called dual sleep approach [59] (Figure 5.16) uses the advantage of using the two extra pull-up and two extra pull-down transistors in sleep mode either in off-state or in on-state. This technique has the advantage of less number of transistors are required for implementing any logic circuit.

5.6.9 Sleepy keeper technique

In this technique low threshold NMOS transistor is put in parallel to a high threshold voltage PMOS sleep transistor in the pull-up network and low threshold voltage PMOS transistor is put in parallel to a high threshold voltage NMOS sleep transistor as

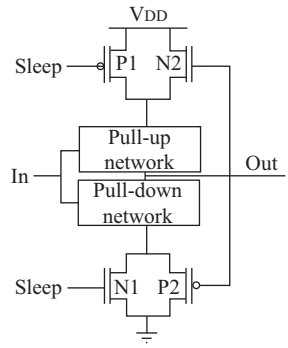


Figure 5.17 Sleepy keeper technique (Source: S. H. Kim, 2006, p.368 [54])

shown in the Figure 5.17. During the active mode, both of the sleep devices and additional MOS transistors are turn-on so they enhanced the logic block's performance. In the standby mode, the sleep transistors are turn-off and hence reducing the leakage current. NMOS transistor passes the logic low voltage completely, while the PMOS transistor passes the logic high voltage completely. In idle mode these two additional transistors preserve the state of the logic block because the NMOS is connected to power supply (logic high) while PMOS is connected to ground (logic low) level [54]. This technique results in increased delay because the output drive strength reduces due to the fact that PMOS device passes a weak low signal and NMOS device passes a weak high signal.

5.6.10 VCLEARIT technique

Figure 5.18 shows the topology of a VCLEARIT (VLSI CMOS LEAkage ReductIon Technique) logic schematic with sleep transistors (P1, P2 and N1) embedded in it. Here three sleep transistors are used for leakage reduction purpose as well as increasing the characteristics of the logic circuit implementation. Two sleep transistors P1 and N1 are of the same threshold voltage as the logic circuit transistors have, while P2 has high threshold voltage. The high threshold voltage transistor is connected between the pull-up and pull-down networks, while other sleep transistors P1, N1 are connected parallel to the pull-up and pull-down circuitry, respectively [60].

In normal operating mode, signal sleep = '1' thus P1 and N1 are turn-off and P2 is turn-on and logic circuitry behaves as the conventional circuitry should. In standby or sleep mode the signal sleep = '0' and hence P1 and N1 are turn-on and P2 is turn-off. So due to breaking of the path between pull-up and pull-down networks, no leakage current should flow. Due to conduction of the sleep transistors P1 and N1, the points X1 and X2 have the same potential as power supply and ground respectively hence no leakage current should flow in pull-up and pull-down circuitry. The output is always zero in standby mode because output is connected to the X2 terminal of the logic circuit which is at ground potential. But leakage loss occurs in P2 transistor because it is connected between different potential terminals. The drawback of this

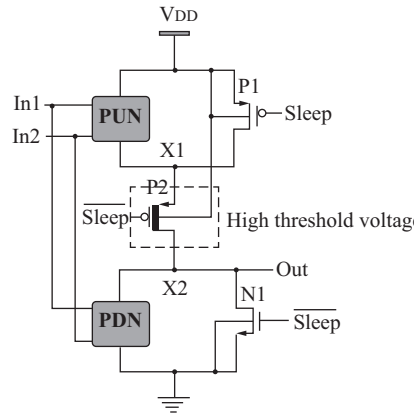


Figure 5.18 VCLEARIT logic schematic (Source: P. Lakshmikanthan, 2007, p.11 [60])

leakage reduction technique is that it requires a controller to automatically control and generate sleep signals to put the circuit in standby mode and also to make active it when required.

5.6.11 GALEOR technique

GALEOR (GAted LEakage transistOR) introduces two additional transistors in the logic circuit to introduce the concept of force stacking which are independent of the input signal combination. Force stack arrangement reduces the leakage current by increasing the resistance of the leakage path. In GALEOR, two gated leakage transistors (GLTs) one NMOS (GLT1) and other PMOS (GLT2) are introduced between the output and pull-up circuitry and the output and pull-down circuitry. These GLTs are having high threshold voltage [61].

GALEOR technique implemented on two input (A and B) NAND gate is shown in the Figure 5.19. In GALEOR technique, a gated leakage NMOS transistor (GLT1) having high threshold voltage is placed between output and pull-up circuit and a gated leakage PMOS transistor (GLT2) having high threshold voltage is placed between output and pull-down circuitry. The concept behind the using high threshold voltage transistors is that it will help in increasing the leakage path resistance so minimize the leakage current.

When the input vector AB = "00", both the NMOS transistors (MN1 and MN2) are turn-off and both the PMOS transistors (MP1 and MP2) are turn-on. This results a voltage closer to the supply voltage at drain terminal of the MN1 transistor, which is able to turn-off the gated transistor (GLT2). This creates a three transistors stack (MN1, MN2 and GLT2) to reduce the leakage current flowing through the circuit. Due to turn-on of the PMOS transistors (MP1 and MP2) they have voltage at their drain

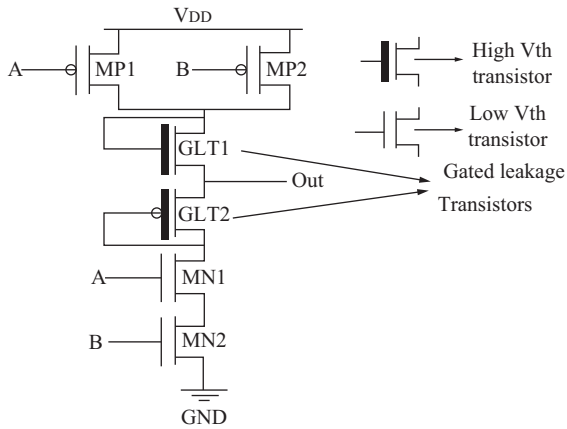


Figure 5.19 GALEOR method for leakage reduction of NAND2 (Source: S. Katrue, 2008, p.575 [61])

terminals which is able to turn-on the gated transistor (GLT1). Therefore, pull-up circuit having negligible on-resistance, this gives full power supply to output node.

Similarly, when input vector AB = "01" or AB = "10", one of the NMOS (MN1 and MN2) and PMOS (MP1 and MP2) transistors are turn-off. This results in a voltage close to the supply voltage at the drain terminal of MN1 to turn-off gated leakage PMOS transistor (GLT2). This creates a two transistor stack to reduce leakage current through the circuit. When input vector AB = "11", both the PMOS transistors (MP1 and MP2) are turn-off. This results in a voltage close to the ground voltage at the drain terminal of MP1 or MP2 to turn-off gated leakage NMOS transistor (GLT1). This creates a two transistor stack in pull-up circuit to reduce leakage current through the circuit. This technique reduces the output voltage swing due to the threshold voltage loss caused by the additional GTLs MOS transistors. This technique suffers a significant problem that is, the low signal is not precisely close to ground and the high signal is not precisely close to V_{DD} [62]. Reduced voltage swing increases the propagation delay through the circuit.

5.7 Leakage analysis

This section explores the importance of different leakage reduction techniques. The results for leakage reduction techniques are elaborated for NAND3 gate for both active and standby modes at different technology nodes. The concluded plots for standby and active mode of power dissipation for NAND3 gate at 65 nm with BPTM technology node for different leakage reduction techniques are shown in Figures 5.20 and 5.21. These results are taken at room temperature with same parameter assumptions whenever required. Power dissipation reduction during standby mode of an application is the important and challenging part of the low power VLSI designer.

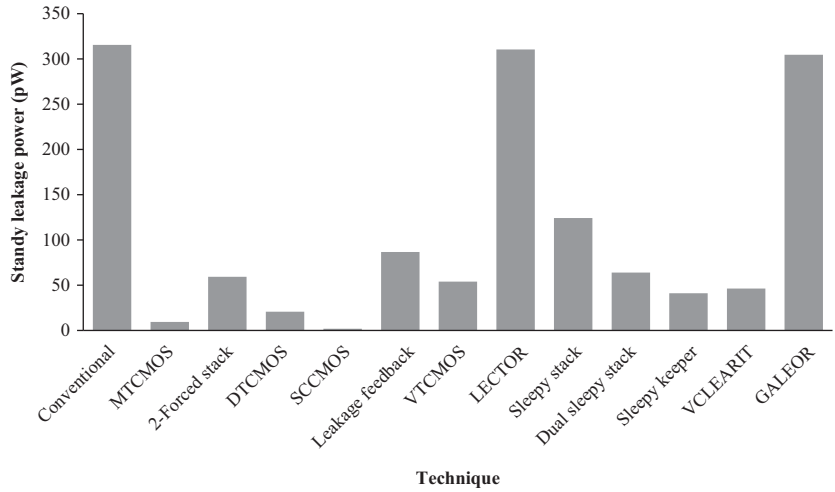


Figure 5.20 Standby power dissipation for NAND3 at 65 nm technology node

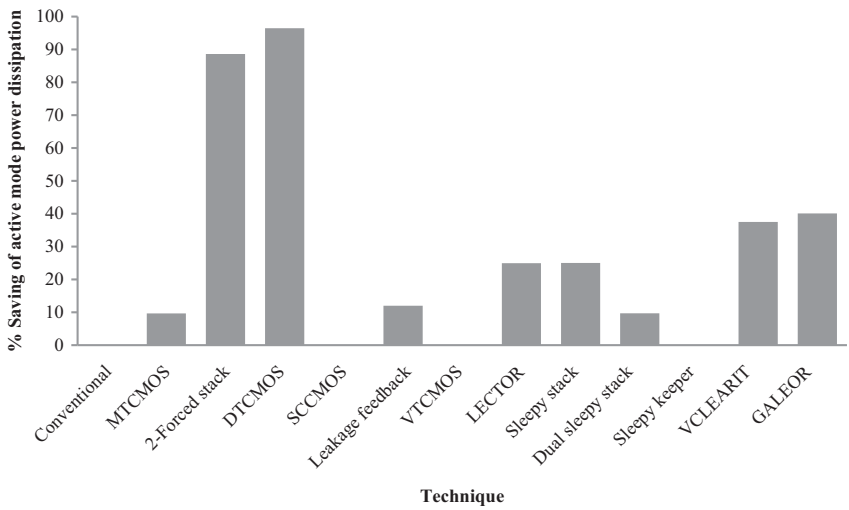


Figure 5.21 Percentage saving of power dissipation in active mode for NAND3 gate at 65 nm technology node

In case of cell phones; the actual usage time is very limited but cell phone is always on. This implies that the standby time of the cell phone is more as compared to active time. It may acceptable to have leakage during the active time but during the standby state it is tremendously wasteful to have leakage, as power is unnecessarily consumed with no useful work being done. Based on the surveyed leakage power reduction techniques we finally got that SCCMOS technique is the best technique for standby

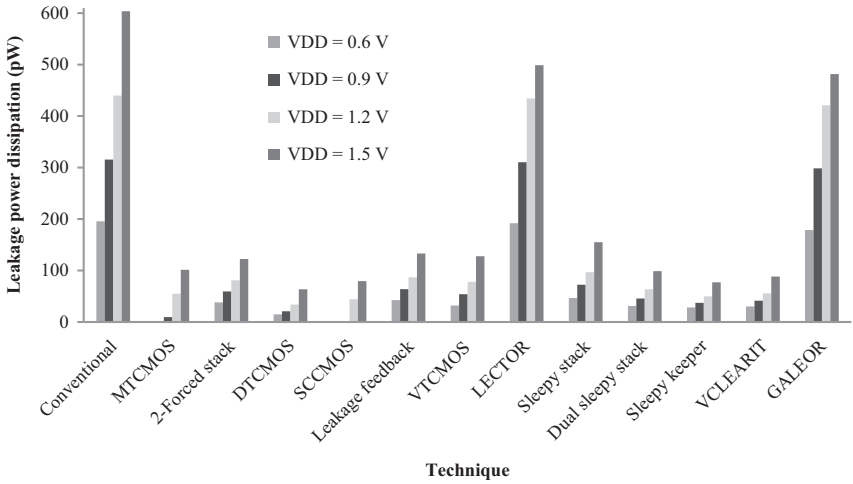


Figure 5.22 Supply voltage variations for NAND3 gate at 65 nm technology node

power dissipation reduction followed by VTCMOS technique. LECTOR technique is less suitable for standby mode power dissipation reduction followed by GALEOR technique.

If the focus is on overall percentage saving of power dissipation in both modes then DTCMOS technique gives the best result followed by 2-forced stack technique. Figure 5.21 shows the percentage saving of power dissipation of different leakage reduction techniques. From this graph we show that sleepy keeper, VTCMOS and SCCMOS techniques have no effect in active mode power dissipation reduction but these are more valuable in standby mode. Actual implementations of DTCMOS technique is more complicated than 2-forced stack technique.

The statistical variations are fetching very vital in current and imminent devices. The key sources of statistical variations are process, voltage and temperature (PVT) variations. PVT variations have become increasing quickly with the continued scaling of ICs and are the major challenge to design low power robust ICs in ultra-DSM regime. These variability issues may lead to significant discrepancies because of improper manufacturing process, biasing and working environmental conditions. PVT variations impact the circuit reliability and leakage power extensively. The variations in physical and electrical parameters of the devices initiate the variability issues. CMOS technology scaling is associated with severe drawbacks when reaching lower nanometer nodes. Leakage current components like gate leakage, sub-threshold leakage and variability issues are increasing drastically with each new lower nanometer node. The supply voltage is the important source of power dissipation as it is directly related to different power dissipation sources. Different power supply voltage values can change the terminal voltages of a MOS device depending on the circuit characteristics and thus responsible to vary the threshold voltage. Change in threshold voltage value affects the leakage power. Power supply voltage variations for different leakage reduction techniques are illustrated in Figure 5.22.

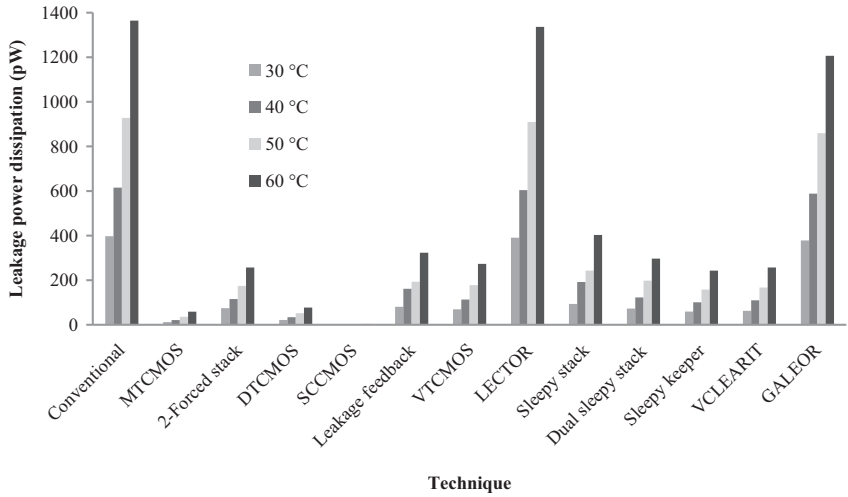


Figure 5.23 Temperature variations for NAND3 gate at 65 nm technology node

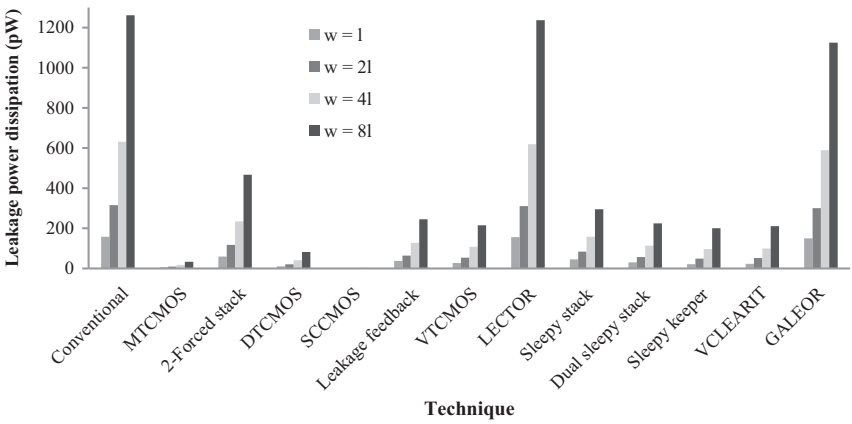


Figure 5.24 Transistor width variations for NAND3 gate at 65 nm technology node

The environmental conditions play a vital role in the process of designing the variations aware ICs. The operating temperature changes the performance of the ICs in several ways. Different parts of a chip have different power densities that result in temperature gradient. Environmental temperature gradient impacts on circuit's parameters. The on-state and off-state currents are varied since ICs feel thermal behavior due to temperature gradient. Temperature variations for different leakage reduction techniques are depicted in Figure 5.23.

Adjustment of the transistor width mitigates the leakage power to meet the required specifications of the systems. Now we consider the effect of transistor width variations for NAND3 circuit for different leakage reduction techniques. A width comparison plot is shown in Figure 5.24.

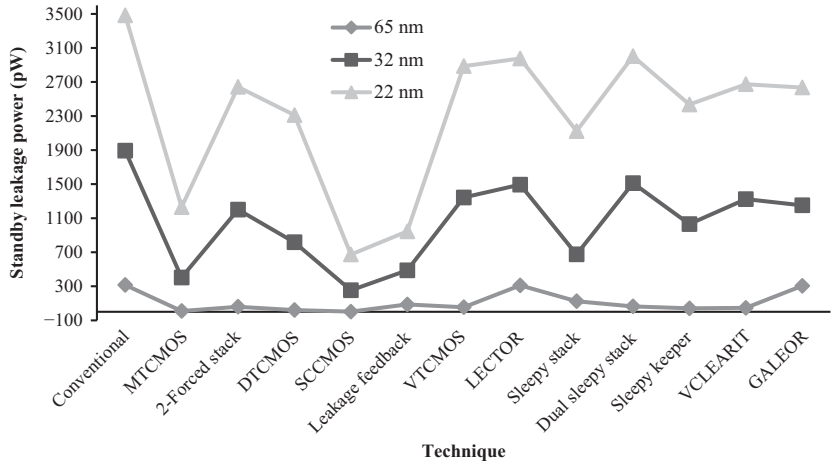


Figure 5.25 Standby leakage power for NAND3 gate for different techniques at 65, 32 and 22 nm technology nodes

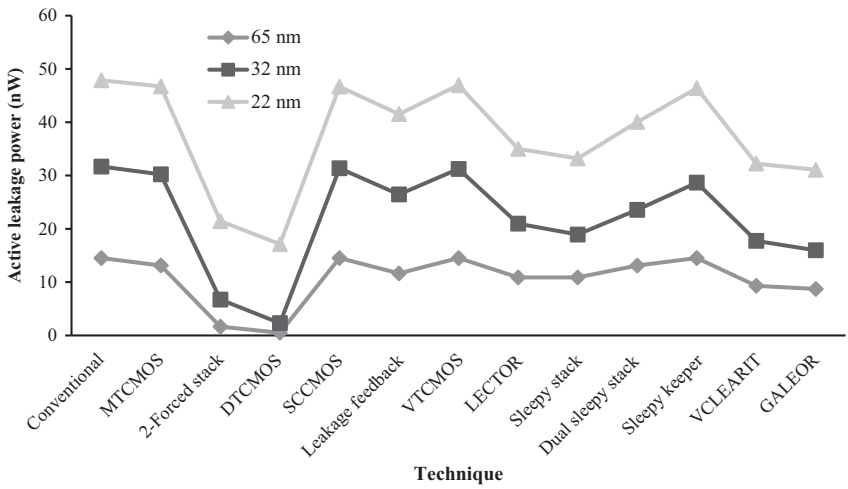


Figure 5.26 Active leakage power for NAND3 gate for different techniques at 65, 32 and 22 nm technology nodes

CMOS technology has continued to scale down at a dramatic rate to opt high performance. Comparative technology scaling effect is shown in Figures 5.25 and 5.26 for various leakage reduction techniques in standby and active modes, respectively. 65, 32 and 22 nm technology nodes are taken as different nodes for different techniques. Power dissipation in both modes is increasing when technology node shifts from higher to lower node.

5.8 Conclusion

Low power VLSI design is attracting a great deal of attention in VLSI design, especially for high performance portable systems. By making the smaller devices, designers can improve the speed of the devices with minimum chip area but scaling of the device increases the leakage power dissipation and variability issues. Currently, minimizing leakage power and variability issues are very demanding area of research. The rapid switching of millions of transistors dissipates tremendous power and overheats the chip, reducing the reliability of the chip and necessitating expensive and large cooling systems. In this chapter we review the various leakage reduction techniques at transistor/circuit/logic level. We explore the merits and demerits of different leakage reduction techniques and show the results in different modes. This chapter is beneficial for quick learning of various leakage reduction techniques. We show the comparative and comprehensive results of the percentage saving of power dissipation in both operating modes of different leakage reduction techniques. SSCMOS technique is the best technique for standby mode while DTCMOS technique is the best technique for overall power dissipation reduction in both modes. Real implementation of SCCMOS and DTCMOS technique is relatively difficult. Based on the surveyed techniques, a designer would be able to select the appropriate leakage optimization technique for a particular level of an application.

References

[1] G. E. Moore, "Cramming more components onto integrated circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998.

[2] K. J. Kuhn, "Moore's law past 32 nm: future challenges in device scaling," 13th *International Workshop on Computational Electronics (IWCE'2009), IEEE*, pp. 1–6, 2009.

[3] R. H. Dennard, F. H. Gaensslen, H.-N. Yu, *et al.*, "Design of ion-implanted MOSFETs with very small physical dimensions," *Proceedings of the IEEE*, vol. 87, no. 4, pp. 668–678, 1999.

[4] T. H. Ning, "A perspective on the theory of MOSFET scaling and its impact," *IEEE Solid-State Circuits Society Newsletter*, vol. 12, no. 1, pp. 27–30, 2007.

[5] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam and D. Burger, "Power challenges may end the multicore era," *Communications of the ACM*, vol. 56, no. 2, pp. 93–102, 2013.

[6] B. Nikolic, "Design in the power-limited scaling regime," *IEEE Transactions on Electron Devices*, vol. 55, no. 1, pp. 71–83, 2008.

[7] R. Vaddi, S. Dasgupta and R. P. Agarwal, "Device and circuit co-design robustness studies in the subthreshold logic for ultralow-power applications for 32 nm CMOS," *IEEE Transactions on Electron Devices*, vol. 57, no, 3, pp. 654–664, 2010.

[8] S. Pandit, C. Mandal and A. Patra, *Nano-scale CMOS Analog Circuits: Models and CAD Techniques for High-level Design*, CRC Press, 2014.

[9] J. M. Rabaey, A. Chandrakasan and B. Nikolic, *Digital Integrated Circuits – A Design Perspective*, Prentice Hall, 2003.
[10] V. Kursun and E. G. Friedman, *Multi-Voltage CMOS Circuit Design*, Wiley, 2006.
[11] H. Stork, "It's all about scale," *IEEE Solid-State Circuits Society Newsletter,* vol. 12, no. 1, pp.33–35, 2007.
[12] M. G. Priya, K. Baskaran and D. Krishnaveni, "Leakage power reduction techniques in deep submicron technologies for VLSI applications," *Procedia Engineering*, vol. 30, pp. 1163–1170, 2012.
[13] D. J. Frank, R. H. Dennard, E. Nowak, *et al.*, "Device scaling limits of Si MOSFETs and their application dependencies," *Proceedings of the IEEE*, vol. 89, no. 3, pp. 259–287, 2001.
[14] B. Nikolic, "Design in the power-limited scaling regime," *IEEE Transactions on Electron Devices*, vol. 55, no. 1, pp. 71–83, 2008.
[15] S. S. B. M. Sallah, H. Mohamed, M. Mamun and M. S. Amin, "CMOS downsizing: Present, past and future," *Journal of Applied Sciences Research*, vol. 8, no. 8, pp. 4138–4146, 2012.
[16] D. J. Frank and Y. Taur, "Design considerations for CMOS near the limits of scaling," *Solid-State Electronics*, vol. 46, no. 3, pp. 315–320, 2002.
[17] A. Morgenshtein, "Short-circuit power reduction by using high-threshold transistors," *Journal of Low Power Electronics and Applications*, vol. 2, no. 1, pp. 69–78, 2012.
[18] N. S. Kim, K. Flautner, D. Blaauw and T. Mudge, "Circuit and microarchitectural techniques for reducing cache leakage power," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 2, pp. 167–184, 2004.
[19] N. Ekekwe and R. Etienne-Cummings, "Power dissipation sources and possible control techniques in ultra-deep submicron CMOS technologies," *Microelectronics Journal*, vol. 37, no. 9, pp. 851–860, 2006.
[20] W. M. Elgharbawy and M. A. Bayoumi, "Leakage sources and possible solutions in nanometer CMOS technologies," *IEEE Circuits and Systems Magazine*, vol. 5, no. 4, pp. 6–17, 2005.
[21] E. N. Shauly, "CMOS leakage and power reduction in transistors and circuits: process and layout considerations," *Journal of Low Power Electronics and Applications*, vol. 2, no. 1, pp. 1–29, 2012.
[22] D. Lee, D. Blaauw and D. Sylvester, "Gate oxide leakage current analysis and reduction for VLSI circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 2, pp. 155–166, 2004.
[23] K. Roy, S. Mukhopadhyay and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, 2003.
[24] X. Yuan, J. E. Park, J. Wang, *et al*., "Gate-induced-drain-leakage current in 45-nm CMOS technology," *IEEE Transactions on Device and Materials Reliability*, vol. 8, no. 3, pp. 501–508, 2008.
[25] M. J. Rabaey, A. Chandrakasan and B. Nikolic, *Digital Integrated Circuits*, Pearson, 2003.

[26] B. C. Paul, A. Agarwal and K. Roy, "Low-power design techniques for scaled technologies," *Integration, the VLSI journal*, vol. 39, no. 2, pp. 64–89, 2006.

[27] S. Jin and Y. Han, "M-IVC: applying multiple input vectors to co-optimize aging and leakage," *Microelectronics Journal*, vol. 43, no. 11, pp. 838–847, 2012.

[28] Z. Abbas and M. Olivieri, "Impact of technology scaling on leakage power in nano-scale bulkCMOS digital standard cells," *Microelectronics Journal*, vol. 45, no. 2, pp. 179–195, 2014.

[29] F. Gong, Y. Shi, H. Yu and L. He, "Variability-Aware Parametric Yield Estimation for Analog/Mixed-Signal Circuits: Concepts, Algorithms and Challenges" *IEEE Design and Test*, vol. 31, no. 4, pp. 6–15, 2014.

[30] C. Shin, "State-of-the-art silicon device miniaturization technology and its challenges" *IEICE Electronics Express*, vol. 11, no. 10, pp. 1–11, 2014.

[31] S. Bobba and I. Hajj, "Maximum leakage power estimation for CMOS circuits" *Proceedings of the IEEE Alessandro Volta Memorial Workshop on Low Power Design*, p. 116, 1999.

[32] S. K. Saha, "Compact MOSFET modeling for process variability-aware VLSI circuit design," *IEEE Access*, vol. 2, no. 2014, pp.104–115, 2014.

[33] M. Lanuzza, F. Frustaci, S. Perri and P. Corsonello, "Design of energy aware adder circuits considering random intra-die process variations," *Journal of Low Power Electronics and Applications*, vol. 1, no. 1, pp. 97–108, 2011.

[34] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," *IEEE Micro*, vol. 25, no. 6, pp. 10–16, 2005.

[35] C. Chiang and J. Kawa, *Design for Manufacturability and Yield for Nano-Scale CMOS*, Springer, 2007

[36] S. Saxena, C. Hess and H. Karbasi, *et al*., "Variation in transistor performance and leakage in nanometer-scale technologies," *IEEE Transactions on Electron Devices*, vol. 55, no. 1, pp. 131–144, 2008.

[37] P. Corsonello, M. Lanuzza and S. Perri, "Gate-level body biasing technique for high-speed sub-threshold CMOS logic gates," *International Journal of Circuit Theory and Applications*, vol. 42, no. 1, pp. 65–70, 2014.

[38] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu and J. Yamada, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, 1995.

[39] Q. Zhou, X. Zhao, Y. Cai and X. Hong, "An MTCMOS technology for low-power physical design," *Integration, the VLSI journal*, vol. 42, pp. 340–345, 2009.

[40] T. Inukai, M. Takamiya, K. Nose, *et al.*, "Boosted gate MOS (BGMOS): device/circuit cooperation scheme to achieve leakage-free giga-scale integration," *IEEE Proceeding of Custom Integrated Circuits Conference*, pp. 409–412, 2000.

[41] L. Wei, K. Roy and V. De, "Low voltage low power VLSI design techniques for deep submicron ICs," *Proceedings of the IEEE International Conference on VLSI Design*, pp. 24–29, 2000.

[42] Z. Liu and V. Kursun, "Charge recycling between virtual power and ground lines for low energy MTCMOS," *8th International Symposium on Quality Electronic Design*, pp. 239–244, 2007.

[43] R. X. Gu and M. I. Elmasry, "Power dissipation analysis and optimization of deep submicron CMOS digital circuits," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 5, pp. 707–713, 1996.

[44] P. Ghafari, M. Anis and M. Elmasry, "Impact of technology scaling on leakage reduction techniques," *IEEE North-East Workshop on Circuits and Systems, NEWCAS*, art. 4488021, pp. 1405–1408, 2007.

[45] L. Yuan and G. Qu, "A combined gate replacement and input vector control approach for leakage current reduction," *IEEE Transactions on VLSI Systems*, vol. 14, no. 2, pp 173–182, 2006.

[46] J. T. Kao and A. P. Chandrakasan, "Dual-threshold voltage techniques for low-power digital circuits," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 7, 2000.

[47] J. Hu, X. Yu and J. Chen, "New low-leakage flip-flops with power-gating scheme for ultra-low power systems," *Information Technology Journal*, vol. 10, no. 11, pp. 2161–2167, 2011.

[48] T. Karnik, Y. Ye, J. Tschanz, *et al*., "Total power optimization by simultaneous dual-Vt allocation and device sizing in high performance microprocessors," *Proceedings of the ACM/IEEE Design Automation Conference*, p. 486, 2002.

[49] A. Valentian and E. Beigné, "Automatic gate biasing of an SCCMOS power switch achieving maximum leakage reduction and lowering leakage current variability," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 7, 2008.

[50] H. Kawaguchi, K. Nose and T. Sakurai, "A super cut-off CMOS (SCCMOS) scheme for 0.5-V supply voltage with pico-ampere stand-by current," *IEEE Journal of Solid State Circuits*, vol. 35, no. 10, 2000.

[51] M. S. Hwang, J. Kim and D. K. Jeong, "Reduction of pump current mismatch in charge-pump PLL," *Electronics Letters*, vo1. 45, pp. 135–136, 2009.

[52] A. Amirabadi, J. Jafari, A. Afzali-Kusha, M. Nourani and A. Khaki-Firooz, "Leakage current reduction by new technique in standby mode," *Proceedings of the ACM Great Lakes Symposium on VLSI*, pp. 158–161, 2004.

[53] J. Kao and A. Chandrakasan, "MTCMOS sequential circuits," *Proceedings of European Solid-State Circuits Conference*, pp 332–335, 2001.

[54] S. H. Kim and V. J. Mooney, "Sleepy keeper: a new approach to low-leakage power VLSI design," *IFIP International Conference on Very Large Scale Integration*, pp. 367–372, 2006.

[55] H. Im, T. Inukai, H. Gomyo, T. Hiramoto and T. Sakurai, "VTCMOS characteristics and its optimum conditions predicted by a compact analytical model," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 11, no. 5, 2003.

[56] W. M. Elgharbawy and M. A. Bayuomi, "Leakage sources and possible solutions in nanometer CMOS technologies," *IEEE Circuits and Systems Magazine*, vol. 5, no. 4, 2005.

[57] N. Hanchate and N. Ranganathan, "LECTOR: a technique for leakage reduction in CMOS circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 2, pp. 196–205, 2004.

[58] J. C. Park and V. J. Mooney, "Sleepy stack leakage reduction," IEEE *Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 11, pp. 1250–1263, 2006.

[59] M. S. Islam, M. S. Nasrin, N. Mansur and N. Tasneem, "Dual stack method: a novel approach to low leakage and speed power product VLSI design," *IEEE International Conference on Electrical and Computer Engineering*, 2010.

[60] P. Lakshmikanthan and A. Nuñez, "VCLEARIT: a VLSI CMOS circuit leakage reduction technique for nanoscale technologies," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 5, pp. 10–16, 2007.

[61] S. Katrue and D. Kudithipudi, "GALEOR: leakage reduction for CMOS circuits," *Proceedings of the 15th IEEE International Conference on Electronics, Circuits and Systems*, art. 4674918, pp. 574–577, 2008.

[62] J. W. Chun and C. Y. R. Chen, "A novel leakage power reduction technique for CMOS circuit design," *International SoC Design Conference*, art. 5682957, pp. 119–122, 2010.

Chapter 6

Thermal effects in carbon nanotube VLSI interconnects

Ashok Srivastava[1] *and K. M. Mohsin*[1]

This chapter is on the thermal effects in contrast to the process variations or leakage issues of the previous chapters. The thermal effect is considered in interconnects of the integrated circuits (ICs). In a paradigm shift, not traditional metal interconnects, but carbon nanotube (CNT) based interconnects are considered. Thus an important post-CMOS era IC design consideration, the CNT-based interconnects is presented in detail in this chapter.

6.1 Introduction

In general interconnect is a metallic wire providing connectivity between two nodes in an integrated circuit (IC) or in a system consisting of many ICs. A node could be any of four terminals of a transistor: source, drain, gate and body or it could be any terminal of a passive components for example, resistors, capacitors and inductors. In very large scale integration (VLSI) circuits, interconnect plays a very important role in determining an overall performance of the system. The overall speed, power consumption and performance depend largely on interconnect technology. To emphasize the importance of interconnect an example from our daily lives might be useful. Imagine very sophisticated computers connected through bad interconnect wires. No matter how sophisticated devices are being used if interconnect performance is not satisfactory it is not possible to achieve the required performance of a system. Therefore, sincere attention should be given to interconnect technology. In this chapter, we briefly discuss the present VLSI interconnect technology and their inherent limitations. After addressing limitations of present technology, we discuss the possible alternative materials. Most of the contents of this chapter will be about carbon nanotube (CNT) based interconnects and their performances. Three common variants of CNT; single wall carbon nanotube (SWCNT), SWCNT bundle, multiwall carbon nanotube (MWCNT) are covered in this chapter. Focus in this chapter will be on to

[1]Division of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, USA

discuss how to integrate electronic properties with thermal properties of all these different CNT-based interconnects in evaluation of their performances. We believe that at the end of the chapter readers will be able to understand why it is important to bring thermal issues in electronics and hopefully they will get the simplest physics-based picture on how to intertwined two different branches of physics in VLSI interconnects. The chapter is organized as follows. In Section 6.2 present status of VLSI interconnect technology is reviewed followed by the review of CNT-based interconnects in Section 6.3. In Sections 6.4 and 6.5, electrical and thermal properties are discussed. High frequency performance is discussed with S-parameters in Section 6.5 followed by conclusion in Section 6.6.

6.2 Present status of VLSI interconnect

In present CMOS technology, mostly copper is used as an interconnect material buried in low-*k* dielectric as shown in Figure 6.1. To reduce capacitive coupling between two adjacent signals interconnect lines, low-*k* dielectric is being used as an interlayer dielectric (ILD) in present CMOS interconnect technology.

A rapid downsizing of MOSFETs has been happening following the Moore's law. In lieu of the continuous downsizing of transistors, interconnect technology also requires to be scaled down to get maximum benefit of downsizing. However, scaling down of Cu/low-*k* interconnect is suffering increased heating, electro-migration and void formation [1, 2]. In scaling of interconnects, low-dimensional effects dominate over the bulk properties of materials. Due to decreased volume, Joule heat generation per unit volume gets increased which causes resistance to increase. Metal ions in interconnect materials are swept by the high electric field and cause void formation which is called as electro-migration. On top of this electro-migration, Joule heating makes things worse. Due to Joule heating center of interconnect reaches melting point

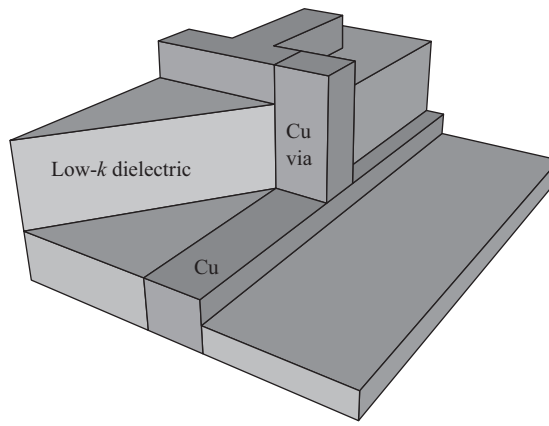


Figure 6.1 Cu/low-k interconnect buried in SiO_2

of the material and thermal run away causes the open circuit. While void and open circuit formation due to electro-migration may happen after longer period of uses, Joule heating may happen right after few clock cycles of use. These reliability issues are the serious bottleneck in scaled CMOS technology nodes. Therefore, researchers are continuously in search of new and novel materials to replace Cu/low-*k* dielectric. Many alternative solutions have been proposed and need materials search including optical interconnects.

6.3 Survey of CNT-based interconnects

After the discovery of CNT in 1991 by Iijima [3] researchers envisioned it as the next-generation interconnect material [4]. Numerous theoretical and experimental researches reached to the conclusion that CNT is the right choice of material due to its excellent electro-thermal properties [5–8]. In Table 6.1, we have summarized few properties relevant to CNT VLSI interconnect technology and compared with that of Cu.

As shown in Table 6.1, different variants of CNT have two orders of more current capacity than Cu, and 5–10 times more thermal conductivity. Due to these exotic properties, researchers explored this carbon-based material not only as the VLSI interconnect [4, 5, 7, 8, 12, 13] but also as sensors [14, 15] and devices [16–19]. In interconnect design, we not only depend on superior electronic properties but also look into thermal properties to avoid Joule heating induced thermal breakdown. Since CNT has high thermal conductivity it can quickly drain out the generated heat into the dielectric. This is why it is perceived that different variants of CNT interconnects will be inherently more thermally stable than the Cu-based interconnect [5, 20]. In explaining electrical properties, Srivastava *et al.* [8, 21] explained how to use one-dimensional fluid-based model for SWCNT and MWCNT interconnects. Single conductor based transmission line model has been proposed by Sarto and SPICE compatible circuit models have been proposed by D' Amore *et al*. [22, 23]. Most of these works highlighted the electronic properties and overlooked the thermal stability. Chiang *et al*. [24] addressed the issue of Joule heating induced performance degrading of Cu/low-*k* interconnects. However, not much is reported in literatures on

Table 6.1 CNT and Cu properties

Properties	**Cu**	**SWCNT**	**MWCNT**
Max current density (A/cm^2)	10^7	$>10^9$	$>10^9$
Melting point (K)	1356	870 [9]	3000–4000*
Thermal conductivity ($Wm^{-1}\,K^{-1}$)	385	1750–6000 [10]	3000 [11]
Mean free path (μm)	0.04	~ 1	25 (100 nm outer diameter)

*Reported in literatures to be close to the melting point of graphite.

Joule heating induced scattering, which is a serious roadblock in achieving the large current density. Pop studied thermal breakdown in metallic SWCNT with Fourier heat equation [9]. One-dimensional Fourier heat equation has been used by Yamada *et al*. [25] and Kitsuki *et al*. [26] to explain experiments of carbon nanofiber thermal breakdown. Further reference literatures will be provided when necessary in the rest of the chapter in discussing all three variants of CNT.

6.4 Electrical properties

A CNT is rolled up single layer graphene sheet consisting sp^2 hybridized carbon atoms with 0.142 nm bond length as shown in Figure 6.2. Depending of the number of layers of graphene CNT could be single wall (SWCNT) or multiwall (MWCNT). Geometry of a SWCNT is just a hollow cylinder, while MWCNT consists of multiple concentric cylindrical shells. Each of these shells is a rolled over single layer sp^2–sp^2 hybridized sheet of carbon atoms. Hence, MWCNT is a piling of concentric multiple SWCNTs where each shell has essentially different diameters. Inter layers are bonded by the weak van der Walls attraction forces with bonding length 0.34 nm [27]. Beside naturally obtained bundle of SWCNT and MWCNT or any of their mixed kind, it is also possible to use isolated SWCNT or MWCNT as VLSI interconnect. In this chapter, we will be focusing only on SWCNT, MWCNT and SWCNT bundle. Same approaches can be applicable for MWCNT bundle and any kind of mixed bundle. In next section, we will be discussing their electrical properties of different CNTs.

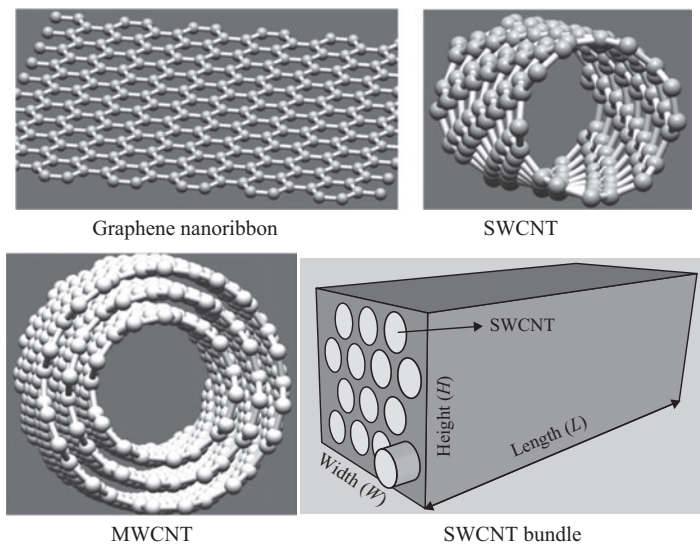


Figure 6.2 Different carbon allotropes: graphene, SWCNT, MWCNT and SWCNT bundle

6.4.1 Equivalent resistance (R_{eqv})

An isolated SWCNT can be implemented in between metal contacts as interconnecting wires as shown in Figure 6.3. Earlier it was very challenging to align SWCNTs in between contacts and recently it has been implemented successfully [28]. For interconnect application SWCNT can also be replaced by MWCNT or SWCNT bundle to get a better performance in terms of current capability. Electrically SWCNT can be approximated as one-dimensional conductor [29]. MWCNT or SWCNT bundle can be approximated as parallel SWCNTs. In case of MWCNT, all the shells are concentric and shell diameters are necessarily different. However, in SWCNT bundle it is not necessary to have different shell diameters. SWCNT bundle shown in Figure 6.2 is of same shell diameter. In this section, we will try to develop a set of general formulae to estimate electrical properties applicable to these three variants of CNTs. From Landauer-Büttiker, formalism dc resistance of a CNT shell can be calculated from (6.1).

$$R_k = \frac{h}{2q^2}\frac{1}{M_k}\left(1 + \frac{L}{\lambda_{eff}(L, D_k, T_k, V)}\right) \tag{6.1}$$

where different parameters are defined as follows:
L = length of interconnect,
D_k = diameter of kth shell in MWCNT or in SWCNT bundle,
T_k = temperature of kth shell,
q = electronic charge,
h = Planck constant and
λ_{eff} = effective mean free path of an electron.

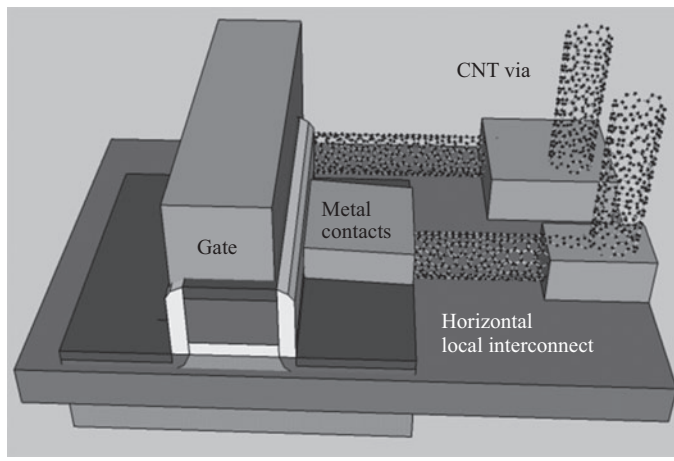


Figure 6.3 SWCNT-based VLSI interconnects

The effective mean free path, λ_{eff}, takes into account of electrons scattering with acoustic and optical phonons and will be discussed in Section 6.4.4. For SWCNT, D_k is just the shell diameter of CNT. First term of (6.1) is quantum in nature and second term is diffusive resistance, which is proportional to the length. Hence, it is apparent from this equation that even though length is almost zero the resistance is nonzero. This nonzero resistance is coming from the effect of contacts and it is quantum in nature. Not to be confused that this first term quantum resistance is not the "contact resistance," it is the contribution of contact in intrinsic part of interconnect. To clarify further for CNT interconnect material while we are taking only CNT resistance in account we are not considering "CNT–metal" interface resistance or "contact resistance." Hence, to get the total resistance including contacts one requires adding contact resistance. Otherwise, this (6.1) will only compute the intrinsic resistance of materials at hand. For SWCNT there is only one shell, however, for MWCNT and SWCNT bundle different shells could be in different temperatures. This is why in (6.1) T_k has been introduced to take into account of the temperature of different shells. The biasing voltage across interconnect is V and M_k is the equivalent conducting spin degenerate channels of interconnect. For metallic SWCNT it is 2. Statistically for naturally obtained SWCNTs, 1/3 are metallic and 2/3 semiconducting. From statistics, we can estimate M_k value of any kind of CNT interconnect with the following equation [30],

$$M_k = \begin{cases} \dfrac{2}{3} & \text{for } D_k < \dfrac{1900}{T_k} \\ a_1 D_k T_k + a_2 & \text{for } D_k > \dfrac{1900}{T_k} \end{cases} \tag{6.2}$$

It is noteworthy that spin degeneracy has been taken into account in above equation. Fitting parameters a_1 and a_2 take the values as $3.26 \times 10^{-4}\,\text{nm}^{-1}\text{K}^{-1}$ and -0.08, respectively. Since (6.2) has been calculated from the knowledge of band structure, it can be used reliably as a compact equation for quick estimation of resistance of interconnect without any detail calculation. Now from (6.1) and (6.2) we know the resistance of each CNT shell regardless of whether they are metallic or semiconducting. Equivalent resistance of MWCNT or SWCNT bundle can be obtained assuming that CNT shells are parallel electrical conductors. On the other hand, SWCNT having only single shell its equivalent resistance R_{eqv} is R_k itself. Details of SWCNT modeling has been explained in the work of Mohsin *et al*. [29]. To estimate equivalent resistance of SWCNT bundle or MWCNT, one requires to have knowledge of total number of CNT shells in that particular kind of CNT-based interconnect. In case of MWCNT, total number of shells can be obtained from (6.3) as follows,

$$N_{shell} = 1 + round\left(\frac{D_{out} - D_{in}}{2\delta}\right) \tag{6.3}$$

where various parameters are described below,
D_{out} = outer diameter of a MWCNT,
D_{in} = inner diameter of a MWCNT and
δ = inter-shell gap and equals to 0.34 nm.

After obtaining total number of shells in MWCNT, equivalent resistance of interconnect can be obtained. Since all these shells are contributing in transport with their average number of conducting channels regardless whether these are metallic or not; by combining them together will give the total contribution of all shells. Equivalent resistance can be then obtained from the following equation [31],

$$R_{eqv} = \left(\frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_k} \cdots + \frac{1}{R_{N_{shell}}}\right)^{-1} = \frac{h}{2q^2}\left(\sum_{k=1}^{N_{shell}} \frac{M_k \lambda_{eff}(L, D_k, T_k, V)}{\lambda_{eff}(L, D_k, T_k, V) + L}\right)^{-1} \tag{6.4}$$

SWCNT bundle can be obtained by piling SWCNTs in different geometrical shapes. In this work, we only considered rectangular shape as shown in Figure 6.2. This is the most common interconnect geometry for VLSI technology. For SWCNT bundle, total number of CNT shells is different and can be obtained from the following equation:

$$N_W = Int\left\{\frac{W - D}{D + d}\right\} + 1 \tag{6.5}$$

$$N_H = Int\left\{\frac{2(H - D)}{\sqrt{3}(D + \delta)}\right\} + 1 \tag{6.6}$$

$$N_{shell} = N_W N_H - Int\left\{\frac{N_H}{2}\right\} \tag{6.7}$$

where various parameters are explained as follows,
N_w = number of CNT shells in the direction of width,
N_H = number of CNT shells in the direction of height,
D = diameter of CNT and
δ = inter shell gap and equals to 0.34 nm.

Function *Int* computes an integer value for everything enclosed by curly brackets. Equation (6.7) counts N_{shell} as the total number of SWCNT shells, which could fit for a rectangular cross-sectional geometry of SWCNT-based interconnect. Once N_{shell} is available, (6.4) can be used to estimate the equivalent resistance of SWCNT–bundle-based interconnect. Again for SWCNT, N_{shell} is 1.

6.4.2 Equivalent inductance (L_{eqv})

After knowing equivalent resistance, it is important to estimate inductance and capacitance of interconnect. For small device dimension besides classical electrostatic capacitance, there is also quantum capacitance. For the same reasoning besides magneto-static inductance, there is also kinetic inductance. In this section, we discuss formulae for both kind of inductances of different kinds of CNT-based interconnect. Magnetic inductance, which depends on the geometrical factor, can be obtained from (6.8) for a cylindrical conductor [8],

$$L_{Mk} = \frac{\mu}{2\pi} \ln\left(\frac{h_k}{2r_k}\right) \tag{6.8}$$

where various parameters are described as follows,
h_k = distance between ground plane and the center of kth CNT shell,
r_k = radius of CNT shell, and
μ = magnetic permeability of CNT.

This magnetic inductance is quite important for SWCNT because of small diameter. However, for a MWCNT with many concentric shells the magnetic inductance plays less important role in overall inductance. Since diameters of outer shells are large, magnetic inductance becomes less significant for outer shells. Again if we count these in parallel to estimate overall inductance it will be less significant. Quantum inductance which comes in series with the magnetic inductance, can be estimated from the following equation [32],

$$L_k = \frac{\pi\hbar}{2q^2 v_F M_k} \tag{6.9}$$

where v_F is Fermi velocity (8.854×10^5 ms^{-1}) [22] and $\hbar$ is reduced Planck constant. To obtain equivalent inductance, one requires adding right sides of (6.8) and (6.9) and then counting them in parallel as done in the case of equivalent resistance in (6.4). Calculations can be easier if we assume that magnetic inductance is less significant in comparison to kinetic inductance. In case of SWCNT bundle, while taking all shells into consideration contribution from magnetic inductance will be divided by total number of by shells. In case of MWCNT, different shells correspond to different diameters. Shell diameters of a MWCNT increases from center to the surface. For increased diameter, magnetic component of inductance becomes less important as described in (6.8). For instance, a MWCNT with 100 nm outer diameter, quantum inductance of innermost shell is 5.8188×10^4 times more than the magnetic inductance of that shell [31]. On the other hand, SWCNT with a diameter of $\sim$1 nm and oxide thickness over which SWCNT is deposited is $\sim$100 Å, the calculated value of $L_{Mk} \sim 1$ pH/μm which is very small compared to the value of L_k which is in the range of nH/μm. With this argument, we can simplify the equivalent inductance with some loss of accuracy and express by the following equation,

$$L_{eqv} = \frac{\pi\hbar}{2q^2 v_F \sum_{k=1}^{N_{shell}} M_k} \tag{6.10}$$

N_{shell} can be calculated for MWCNT and for SWCNT bundles from (6.3) and (6.7), respectively. In case of metallic SWCNT, total number of conducting channels are only 2. For SWCNT bundle- and MWCNT-based interconnect we need to carry summation over all CNT shells to obtain total number of conducting channels. It is to be noted that total number of shells is different than the total number of conducting channels. Number of shells is a physical quantity which counts total number of SWCNT tubes fitted in to the interconnect geometry. On the other hand, number of conducting channels are electronic channels, which are coming from the band structure calculations. One CNT shell might have multiple electronic channels for transportation of electrons. Finally for SWCNT bundle-based interconnect assuming same diameters of all CNTs we can further simplify the equivalent inductance from (6.2) and (6.10) as follows,

$$L_{eqv} = \frac{\pi\hbar}{2q^2 v_F \sum_{k=1}^{N_{shell}} M_k} = \frac{3\pi\hbar}{4q^2 v_F N_{shell}} \tag{6.11}$$

6.4.3 Equivalent capacitance (C_{eqv})

As in inductance, small dimensional interconnects have two kinds of capacitances. One is electrostatic in nature and depends on the geometric shape and dielectric constant of materials. The other one is quantum in nature. Electrostatic capacitance of a CNT is similar to a cylindrical conductor and can be estimated as follows,

$$C_{Ek} = \frac{2\pi\varepsilon}{\ln \frac{h_k}{2r_k}} \tag{6.12}$$

where ε is dielectric permittivity, h_k is distance between ground plane and the center of k^{th} CNT shell and r_k is radius of CNT shell.

Quantum capacitance is in series with this electrostatic capacitance and can be estimated from the following equation,

$$C_Q = \frac{2q^2}{h v_F} M_k \tag{6.13}$$

For metallic SWCNT M_k is 2. Usually, we want a material with low dielectric constant as an ILD to burry interconnect into it. Since electrostatic and quantum capacitances are in series, the one lower in value will dominate in estimation of overall capacitance. If the dielectric constant is low and the distance of interconnect layer from the ground plane is high electrostatic capacitance will dominate over the quantum capacitance. Actually, there is no general rule to find out which one will dominate over the other one. Therefore, for detail calculations one always requires to include both of these to estimate equivalent capacitance numerically. However, to express analytically we can make simplified assumptions. Electrostatic capacitance

is dominant over quantum capacitance only for MWCNT with fewer shells. For a MWCNT with more shells quantum capacitance is approximately thousand times more than the electrostatic capacitance of the outermost shell [31]. Hence, in calculation of equivalent capacitance considering only quantum capacitance is a good approximation. Considering only quantum capacitance, equivalent capacitance can be then estimated from the following equation,

$$C_{eqv} = \frac{2q^2}{hv_F} \sum_{k}^{N_{shell}} M_k \tag{6.14}$$

For SWCNT bundle, it can be further approximated using (6.2) as follows,

$$C_{eqv} = \frac{2q^2}{hv_F} \sum_{k}^{N_{shell}} M_k = \frac{4q^2 N_{shell}}{3hv_F}. \tag{6.15}$$

6.4.4 Effective mean free path (λ_{eff})

One of the most important parameter in electronic transport properties is the carrier mean free path. In CNT, electrons are the major charge carriers. Therefore, in this section we will discuss electrons with various mean free paths associated with different collisions. Finally, Matthiessen's Rule will be used to estimate effective mean free path (λ_{eff}) of an electron. Effective mean free path is the distance an electron can travel before it get scattered. An electron can get scattered due to collision with another electron or phonons. Phonons are quantized lattice atom vibrations. Due to vibrations, atoms are displaced from their equilibrium position and this changes the potential profile in the atomic scale. This change in potential is the cause for electrons to get scattered. This is the simple picture how electrons get scattered by phonons. Mostly phonons are of two different types, optical phonons and acoustic phonons. Both of these kinds of phonons interact with mobile electrons and thus scatter electrons.

Also electrons spontaneously get scattered with optical and acoustic phonons. Scattering length of electrons due to acoustic (λ_{ac}) and optical phonons can be estimated from following equations [33]:

$$\lambda_{ac} = \frac{400.46 * 10^3 D}{T} \tag{6.16}$$

$$\lambda_{op} = 56.4D \tag{6.17}$$

Here D is the diameter of SWCNT and T is the temperature of the shell. For large diameters electrons get more space to travel before scattered by an acoustic phonon. On the other hand, increase in temperature increases population of acoustic phonons, which eventually increases the number of collisions. Hence, increase in temperature decreases the electrons scattering length. Scattering length due to

optical phonon is directly proportional to the diameter of CNT as shown by (6.17). In these above events, no phonon absorption and emission by electrons was involved. Only spontaneous scattering due to acoustic and optical phonon was involved. However, an electron can get scattered by absorbing or emitting an optical phonon. Since acoustic phonon lies in low frequency bands of vibrational modes these are not absorbed or emitted by electrons. Therefore, two more scattering events related to emission and absorption of optical phonon need to be discussed. The scattering length due to optical phonon absorption, $\lambda_{op,abs}$ has been modeled by the following equation [9],

$$\lambda_{op,abs} = \lambda_{op} \frac{N_{op}(300) + 1}{N_{op}(T)} \tag{6.18}$$

Here λ_{op} can be obtained from (6.18) and N_{op} describes the optical phonon occupation, which can be calculated from Bose-Einstein statistics.

$$N_{op} = \frac{1}{\exp\left(\frac{\hbar\omega_{op}}{K_B T}\right) - 1} \tag{6.19}$$

Here ω_{op} is the optical phonon frequency and its typical energy value varies from 0.16 eV to 0.20 eV. For the sake of numerical calculation, one can take a value of 0.16 eV. From (6.19), it is obvious that as temperature increases N_{op} increases. Consequently if N_{op} increases, scattering length due to optical phonon absorption ($\lambda_{op,abs}$) decreases according to (6.18). Therefore, in high temperature CNT interconnect, electron suffers more scattering due to optical phonon absorption. Electrons also get scattered due to emission of optical phonons. Optical phonon emission process has two components, one is for the absorbed energy and another is the electric field induced due to bias across the SWCNT length. Both of these components can be estimated as follows,

$$\lambda_{op,ems}^{abs} = \lambda_{op,abs} + \frac{N_{op}(300) + 1}{N_{op}(T) + 1} \lambda_{op} \tag{6.20}$$

$$\lambda_{op,ems}^{fld} = \frac{\hbar\omega_{op} - K_B T}{q\frac{V}{L}} + \frac{N_{op}(300) + 1}{N_{op}(T) + 1} \lambda_{op} \tag{6.21}$$

Here q is electronic charge, V is the bias voltage across CNT and L is length. From Matthiessen's Rule one can estimate the mean free path due to these above-mentioned scattering processes and is expressed as follows,

$$\frac{1}{\lambda_{op,ems}} = \frac{1}{\lambda_{op,ems}^{abs}} + \frac{1}{\lambda_{op,ems}^{fld}} \tag{6.22}$$

Equation (6.22) estimates the effective mean free path only due to optical phonon emission. Following equation can be used to estimate overall mean free path of electrons.

$$\frac{1}{\lambda_{eff}} = \frac{1}{\lambda_{ac}} + \frac{1}{\lambda_{op,ems}} + \frac{1}{\lambda_{op,abs}} \tag{6.23}$$

Equation (6.23) is a closed form of estimating effective mean free path of an electron. More advanced methods based on first principles can be used which is computationally very expensive and not usable in analytic form. This is why (6.23) can serve for quick estimation of the performance analysis of CNT interconnects. This effective mean free path (λ_{eff}) can be used in (6.4) to estimate temperature and geometry-dependent equivalent resistance of CNT interconnects.

6.4.5 Equivalent circuit

Once we know the equivalent resistance, inductance and capacitance of a CNT interconnect, we can model its equivalent circuit. Yao *et al*. and D'Amoro *et al*. [34] have showed how to model equivalent circuit for MWCNTs [21]. Electrical equivalent circuit for SWCNT bundle has been modeled by Sarto *et al*. [35]. These models mostly involved RLC parameters of multiple lines and coupling impedance in between lines. In inductance and capacitance, they considered both the classical and quantum counterparts. However, it is required to have a single conductor transmission line model to describe the electrical performances in a way that is more compact. It is to be noted that compact modeling can help to simulate a large system with limited computing resource. Sarto *et al*. [22] proposed single conductor transmission line model for the MWCNT which can be modified for any kind of CNT. Following this one can have single conductor transmission line model for any kind of CNT as follows. Only the equivalent *R*, *L* and *C* will be different depending on the kind of CNT interconnect needs to be modeled. For high frequency characterization of interconnect, this transmission line model will be very useful. In scattering parameter calculations, this equivalent single conductor model has been used along with per unit length circuit parameters. Equivalent circuit is as shown in Figure 6.4.

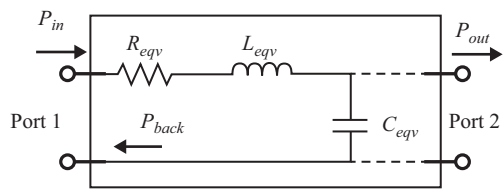


Figure 6.4 Single conductor transmission line model of CNT interconnect

6.5 Thermal properties

Knowing electronic parameters of CNTs, it is now important to know thermal properties and to study how these affect electronic transport. Just to remind the motivation of this chapter again, Joule heating limits the transport of CNT VLSI interconnects. Therefore, it is important to analyze different CNT variants in terms of thermal stability. The fundamental approach is to model temperature distribution and to study thermal stability of interconnect. Phenomenological heat diffusion equation can be used as long as the length of CNT is long enough in comparison to the mean free path of a phonon. If the CNT length becomes comparable to the phonon mean free path necessary physics cannot be described by the phenomenological Fourier heat equation. In later case, one needs to solve the Boltzmann transport equation (BTE) for phonons which is less intuitive and computationally more expensive. Again standard Boltzmann equation has its own limitation of treating phonon as a classical particle instead of quantum nature of phonon. Cahill *et al*. [36, 37] have reviewed recent progress of thermal transport in nanoscale. In most of present practical purposes CNT lengths are of few micrometers, whereas phonon effective mean free path is few hundreds of nanometer. Therefore, it is permissible to use Fourier heat equation with necessary boundary and initial conditions. In this section, we will describe different Fourier heat equation with their boundary conditions for different kind of CNT interconnects. Since MWCNT and SWCNT bundle consist of SWCNT in different geometric orientations it is natural to study the thermal properties of SWCNT-based interconnect first and then other variants of CNT.

6.5.1 Thermal properties of SWCNTs

SWCNT being quasi one-dimensional and does not have enough surface area and cross-sectional area to dissipate heat. Hence it is prone to thermal breakdown more than any other variants of CNT. Even though SWCNT is highly heat conductive, it has been observed experimentally that the conducting CNT breaks down due to Joule heating and thus limits its current density [38, 39]. Thermal breakdown of SWCNT has been studied by Pop *et al*. [9]. Huang *et al*. [40] have studied thermal transport and observed experimentally that the hottest spot is located at the center of the tube from where breakdown is initiated. Geometry of SWCNT is essentially a single one-dimensional wire. So its Fourier heat equation can be described by one-dimensional equation.

$$\frac{\partial}{\partial z}\left(A\kappa\frac{\partial T}{\partial z}\right)+p=A\rho c\frac{\partial T}{\partial t} \tag{6.24}$$

In (6.24), κ is the heat conductivity, A is the cross–sectional area ($\pi \times d \times t$), T is the temperature at a given point of SWCNT, p is Joule heating source power per unit length, c is the specific heat of CNT and ρ is the density of CNT. A typical diameter of SWCNT is 1 nm and thickness 0.34 nm which is the interlayer distance

in graphite. Under the assumption of uniform heating, uniform cross-sectional area of CNT, steady state solution can be obtained using the work of Pop *et al*. [9],

$$A\frac{\partial}{\partial z}\left(\kappa\frac{\partial T}{\partial z}\right)+p-g\left(T-T'\right)=0 \tag{6.25}$$

where T' is the substrate temperature, g is the measure of heat conductivity of CNT through substrate. The better g will be, CNT will be more thermally stable. Right hand side of (6.25) equals zero since steady state solution is sought for. Being one-dimensional problem, only one boundary condition is required to solve this problem. Typical substrate temperature is $60 \sim 70°$C. Again g depends on the interface of CNT and substrate and it is to be measured experimentally or has to be estimated from computationally very extensive first principle calculations. Pop *et al*. [9] measured for silicon substrate and found its value as $0.15\,\mathrm{Wm^{-1}\,K^{-1}}$. Thermal conductivity is the most disputed parameter for CNT. It ranges from few thousand to few hundreds of $\mathrm{Wm^{-1}\,K^{-1}}$. In the work of Yamada *et al*. [25] and Kitsuki *et al*. [26] thermal transport has been studied considering the thermal conductivity of CNT as constant. However, studies in References 9, 20 and 41–44 have been shown that the thermal conductivity is temperature dependent. This is why one always needs to check with the most recent agreed experimental values for thermal conductivity until an established theoretical and experimental value have been obtained. For a constant thermal conductivity a compact analytical solution has been shown in Reference 9 as follows,

$$T(z)=T'+\frac{p}{g}\left[1-\frac{\cosh\left(\dfrac{z}{\sqrt{\kappa Ag}}\right)}{\cosh\left(\dfrac{L}{2\sqrt{\kappa Ag}}\right)}\right] \tag{6.26}$$

One of the important outcomes of Pop's study [9] is defining thermal healing length as follows,

$$L_H=\sqrt{\kappa Ag} \tag{6.27}$$

If the length of CNT is very large in comparison to the thermal healing length (L_H) most of the heat will be lost by the substrate. On the other hand, if the length is comparable to L_H most of the heat diffuses through the contacts. In (6.26), p/g determines the peak temperature of the hottest spot which is the midpoint temperature of SWCNT. Now from (6.26), one can find the temperature distribution for a given Joule heating (p) per unit length which is I^2R per unit length. Therefore, (6.26) is resistance (R) dependent. Again from (6.4) we know that R is T dependent. Hence, there is a nonlinear relation between R and T. This nonlinear relation can be taken into account by the iterative scheme, which is discussed in Section 6.5.4. Once T and R can be calculated from the coupled electro-thermal equations, thermal stability can be determined. If the temperature at any point of a SWCNT goes above the breakdown temperature 873 K, it melts down at that point and breaks the circuit. Since peak

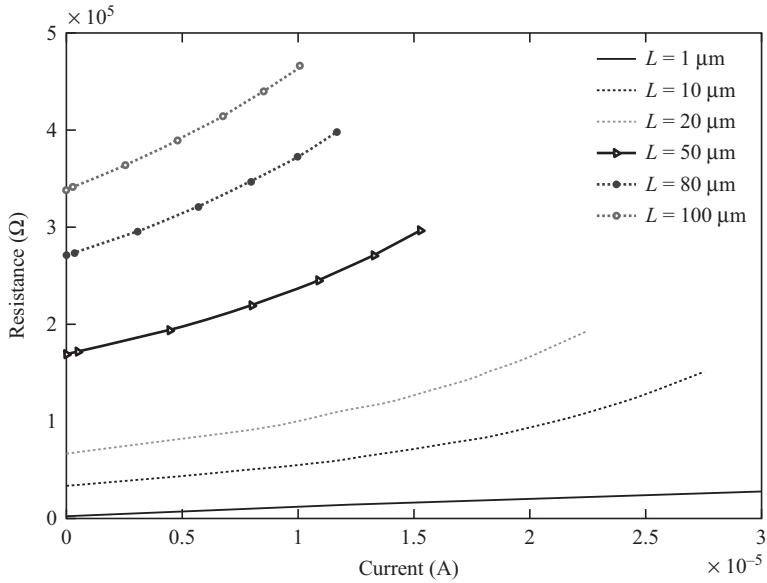


Figure 6.5 Length dependence resistance of SWCNT interconnect

temperature is being determined by p ($= I^2R$), thermal stability depends directly on the bias current. This is how the Joule heating limits the current density and in worst case, causes thermal breakdown of interconnects. For a 3 μm long SWCNT and 2 nm diameter, maximum current has been calculated to be 20 μA [9, 39]. Since longer SWCNT has more contacts with substrate it can dissipate heat quicker than the shorter one. Hence a length dependence of thermal stability comes into the picture. In the work of Mohsin *et al*. [29], length dependence of resistance has been presented as shown in Figure 6.5. To summarize, length, diameter, temperature and bias voltage or bias current decide the mean free path and resistance of CNT interconnect. Resistance and temperature are intertwined with each other and need to be solved with interactive scheme described in Section 6.5.4.

6.5.2 Thermal properties of SWCNT bundle

Heat equation for SWCNT bundle is a three-dimensional problem to be solved with necessary boundary conditions. If the cross-sectional area is very small and the length is very large, we can assume it as one-dimensional problem. For one-dimensional case, we already discussed the solution of heat equation in previous section. However, in case of local interconnect the cross-section is large enough and interconnect is not comparatively long enough to assume one-dimensional problem. In this case, thermal equilibrium is possible over the length but not over the cross-section. Hayashi *et al*. [45] noted that there is anisotropy of heat conductivity in MWCNT. This

anisotropy suggests that heat conductivity is very high in axial direction in comparison to the radial direction. Since MWCNT and SWCNT bundle only vary in placement of SWCNT shell in different geometric orientation; same anisotropy is also possible for SWCNT bundle. Heat conductivity will be always dominant in axial direction while electron and phonon transport are happening in the shell. In contrast whenever transport involves with an adjacent shell, heat carriers need to face the inter-shell thermal resistance. Hence heat conductivity will not be high in the radial direction. Therefore, in cross-section of SWCNT-based interconnect temperature gradient will be observed. There will be a temperature distribution in the cross-section of SWCNT-bundle-based interconnect due to this anisotropy of heat conductivity. A cross-sectional distribution of temperature means different SWCNT wires will be at different temperatures causing different temperature-dependent resistance (R_k). Thus a three-dimensional heat equation is to be solved as follows,

$$\frac{\partial}{\partial x}\left(A\kappa_x\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(A\kappa_y\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(A\kappa_z\frac{\partial T}{\partial z}\right)+p-g\left(T-T'\right)=A\rho c\frac{\partial T}{\partial t} \tag{6.28}$$

where κ_x, κ_y and κ_z are heat conductivity in x, y and z directions. The geometry of SWCNT-bundle-based interconnect is a three-dimensional rectangular bar, $p = (I^2R_{eqv})/(WxLxH)$, where W, L and H are width, length and height of interconnect. Substrate temperature T' can be taken as 300 K and will serve as a boundary condition for solving (6.28). Definition of other parameters are given in the previous section. Under the following assumption, one can simplify (6.28) to solve this problem analytically or numerically. Following assumptions are made:

1. Heat generation is uniform throughout the whole interconnect.
2. Cross-section of interconnect is uniform.
3. Anisotropy of heat conductivity is only two dimensional: radial and axial direction of a CNT.
4. Steady state solution.
5. Interconnect length is short enough so there will be no temperature variation along the length (z-axis).

Considering above-mentioned assumptions, we can have the following equation to solve,

$$\kappa_x\frac{\partial^2 T}{\partial x^2}+\kappa_y\frac{\partial^2 T}{\partial y^2}+p-g\left(T-T'\right)=0 \tag{6.29}$$

In (6.29), constant cross-sectional area (A) can be absorbed into other constants. Heat conductivity through substrate (g) is yet to be measured experimentally for SWCNT-bundle-based interconnect.

6.5.3 Thermal properties of MWCNT

The MWCNT is inherently more stable than SWCNT in terms of thermal stability. On top of thermal stability, growth techniques of MWCNT are easier than SWCNT. This is why MWCNT is preferred over SWCNT as the VLSI interconnect. For better understanding the performance of various CNT interconnects and to compare them it is essential to study MWCNT interconnect as well. Temperature distribution along the MWCNT considering Joule heating has been reported by Feng *et al*. [46] and later they studied the cross-talk effects for the VLSI interconnect [47]. In most of early studies of MWCNT-based interconnect, radial heat flow has been neglected assuming graphite like isotropic thermal conductivity of MWCNT [25, 46]. Hayashi *et al*. [45] have reported that due to anisotropy of thermal conductivity temperature variation can also be observed in the cross-section of MWCNT. It is necessary to include this anisotropy in consideration to get accurate results. From the experimental results Hayashi *et al*. [45] calculated thermal conductivity in axial direction as $\kappa_{axial} = 1800\,\mathrm{Wm^{-1}\,K^{-1}}$ and in radial direction, $\kappa_{radial} = 0.05\,\mathrm{Wm^{-1}\,K^{-1}}$. This is a significant anisotropy to be taken into account in thermal study. Since axial component is very high in comparison with the radial component it is suggestive that heat dissipation along the length of MWCNT is quicker than in radial direction. Hence MWCNT interconnect will quickly reach thermal equilibrium along all over the length but in cross-section it may not be that quicker. This is how anisotropy of thermal conductivity will cause temperature variation in cross-section. However, temperature variation will not be significant in the direction of length for aforementioned reason. One can take this as an advantage to reduce the problem from three dimension to two dimension by considering only the cross-section. Again under the assumption of uniform diameter throughout and all over the length, one can assume that heat will not flow in circumferential direction too. Therefore, one can reduce the heat equation to a radial equation. Although, Hayashi *et al*. [45] modeled MWCNT in two-dimensional cylindrical co-ordinates with anisotropic values of thermal conductivity they did not took Joule heating generation term in their governing equation. In the work of Mohsin *et al*. [31], heat generation term has been considered but conduction through dielectric (g in (6.30)) has not been presented. To be more accurate one should always include all heat sources and sinks which are shown in Figure 6.6.

For giving an example here, we have adopted the work from Reference 31 here in. Governing equation to be solved in for MWCNT is as follows,

$$\kappa_{axial}\frac{\partial^2 T}{\partial z^2} + \kappa_{radial}\frac{1}{r}\left\{r\frac{\partial^2 T}{\partial r^2} + \frac{\partial T}{\partial r}\right\} + p = 0 \tag{6.30}$$

Here thermal conductivity in axial direction is expressed through as $\kappa_{axial} = 1800\,\mathrm{Wm^{-1}\,K^{-1}}$ and in radial direction through $\kappa_{radial} = 0.05\,\mathrm{Wm^{-1}\,K^{-1}}$. Joule heat generation term (p) is V^2/R per unit volume. Equation (6.30) gives steady state solution for temperature inside the cross-section and along length (z-axis) of MWCNT interconnects. As boundary condition for MWCNT, outer CNT shell is in thermal equilibrium with the dielectric. In addition, two ends of interconnects are also in thermal equilibrium with the ambient chip temperature.

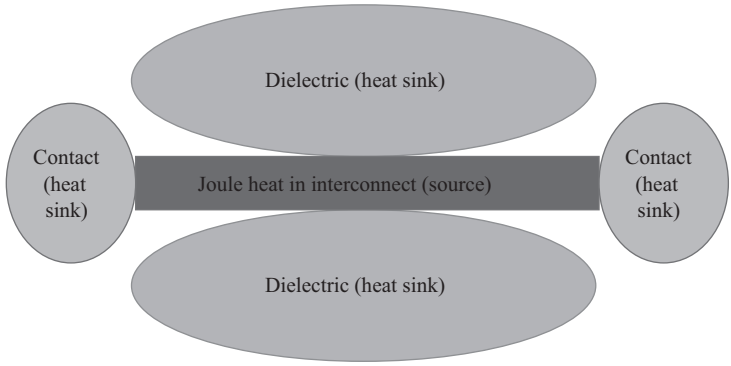


Figure 6.6 Heat source and sinks in VLSI interconnect

6.5.4 Iterative scheme for R and T

Since resistance depends on temperature (T) and temperature depends on resistance (R) via Joule heat (I^2R); there is a nonlinear relation between R and T. This nonlinear relation requires self-consistent solution for R and T iteratively. Electrical transport equation tells us the temperature dependence of resistance. On the other hand, heat equation tells us about the resistance dependency on temperature. In the beginning of iteration, it can be assumed that whole simulation domain is in thermal equilibrium with the ambient temperature of the chip. In many studies, it has been taken as room temperature. However, temperature can be elevated due to the operation of other circuitry around a particular block of interconnect. Usually 350 K is a good approximation, which can be taken as high ambient temperature of silicon chip. Using room temperature 300 K is not a good assumption to start with since normal chip operation is above room temperature. Once bias either current or voltage is applied, interconnect will develop Joule heat and will not be in thermal equilibrium with the surroundings any more. Flow chart in Figure 6.7 presents steps for achieving self-consistent temperature and resistance for the solution.

First step is to assume suitable initial temperature which is ambient temperature of interconnect under study. After knowing the temperature one can calculate the mean free path of electron (λ) at that temperature using (6.24) described in Section 6.4.3. By using (6.4) equivalent resistance (R_{eqv}) can be calculated. After calculating the equivalent resistance, Joule heat can be calculated by multiplying with current squared. Then using Joule heat term in any of governing heat equations depending on the type of CNT interconnect, temperature distribution inside interconnect can be obtained. These governing equations are described in Sections 6.5.1–6.5.3. If the newly calculated temperature is not within the tolerance limit one should repeat the loop starting with this newly calculated temperature. In our earlier work [31], tolerance has been taken as 0.01 K. Once the difference of newly calculated temperature is within the tolerance limit one should stop this calculation. Again if the temperature reaches above 873 K, CNT will breakdown from the middle. Iteration should also

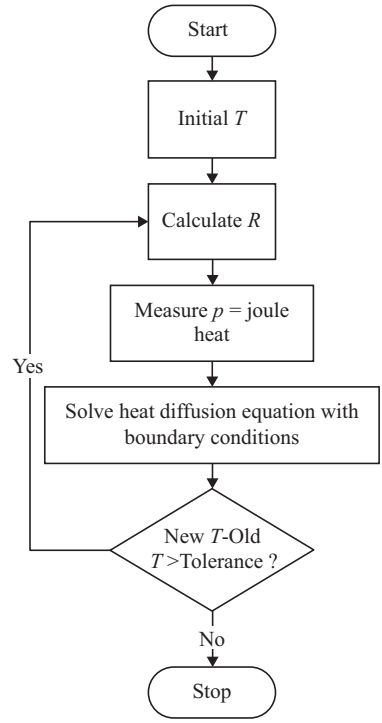


Figure 6.7 Iterative scheme for electro-thermal coupled equations

stop at this melting point denoting CNT's breakdown. The breakdown resistance and current are then recorded. That breakdown resistance is the highest resistance before breakdown occurs.

6.5.5 Temperature profiling inside the interconnect

In SWCNT, temperature profiling in one dimensional is showing temperature variation only in the direction of length of the CNT. It has been experimentally verified that the hot spot is at the center of the CNT and Joule heating induced breaking occurs at the center of CNT. In Figure 6.8 an example of SWCNT temperature distribution is presented.

From Figure 6.8, it is seen that the maximum temperature depends on the biasing voltage and is peaked at the center. SWCNT wire of 1 nm diameter will survive any voltage below 4 V. Above 4 V, probability of melting down from the center is high and thermal breakdown resistance become infinity which can be observed in Figure 6.5. In Table 6.2, we have shown how resistance varies after considering Joule heating.

In Figure 6.9, cross-sectional temperature profile of a SWCNT-bundle-based interconnect has been shown in Reference 50. The central CNTs is mostly at high temperature and center most shell achieve the maximum temperature. Therefore, if

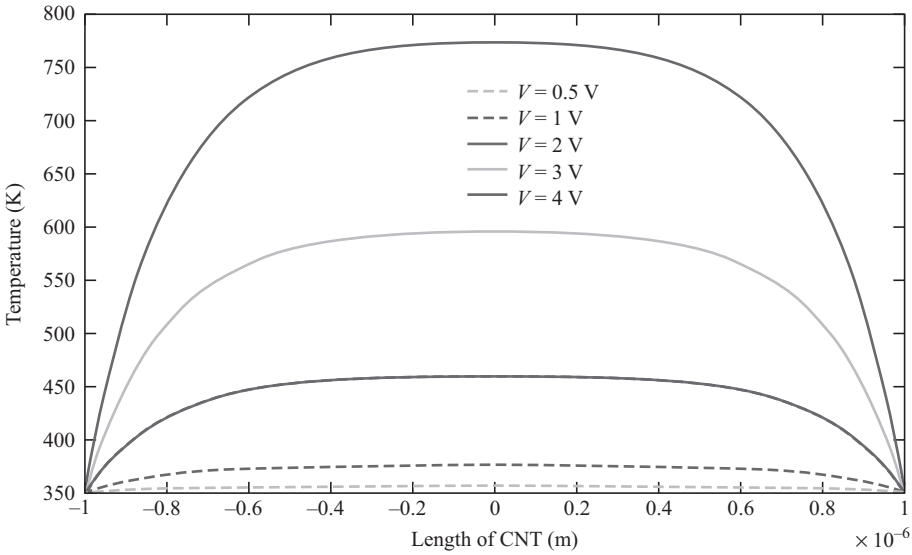


Figure 6.8 Temperature profile of SWCNT interconnects

Table 6.2 Comparison of Joule heating based modeling with experimental measurements [48, 49] and other theoretical models [8, 31] for SWCNT

References	Length (μm)	D_{in} (nm)	D_{out}(nm)	R (kΩ)	With Joule heating [31] (kΩ)
Nihei *et al*. [48]	2	3.88	10	1.60	2.257
Srivastava and Yao [8]				1.90	
Li *et al*. [49]	25	50	100	0.035	0.02781
Srivastava and Yao [8]				0.042	

any breakdown occurs it will occur from the center of the central CNT shell. In Table 6.3, equivalent resistance is presented for SWCNT bundle interconnect for different bias currents.

For MWCNT the temperature distribution has been calculated in Reference 31 and shown in Figure 6.10. It has been found that MWCNT is inherently thermally more stable than any other kind of CNT interconnects.

Most theoretical models assume that one third of the MWCNT shells are metallic and rest of them are semiconducting which is statically sound. On the other hand, in real measurement this statistical assumption will not work. In real measurement, one could get different number of metallic shells than the theoretical assumption. This is why total resistance can vary from experiment to experiment.

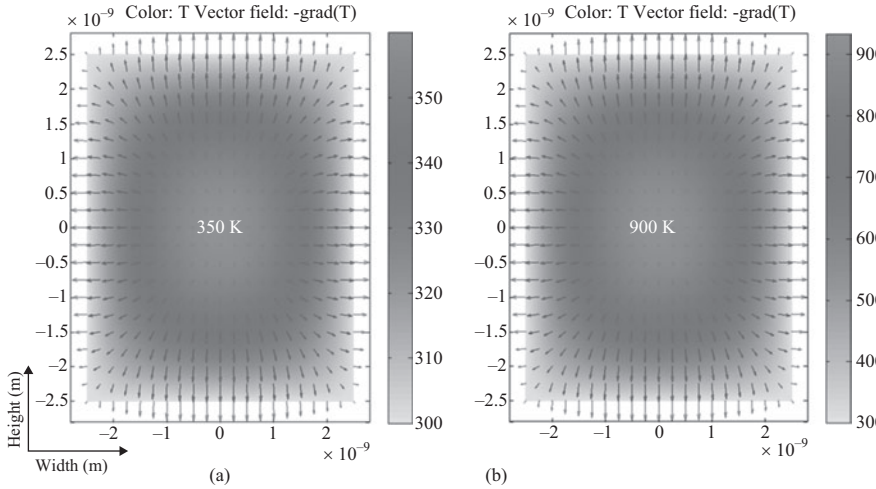


Figure 6.9 Cross-sectional temperature distributions and heat flow vector (arrow) for (a) 10 mA, D = 1 nm, L = 1 μm, (b) 10 mA, D = 4 nm, L = 1 μm. In case of (b) central shell temperature is above melting point (873 K), therefore breakdown will occur

Table 6.3 Temperatures and equivalent resistances of SWCNT bundle interconnect at various bias currents

Bias current (mA)	Width = height (nm) $D = 1$ nm unless mentioned	Number of SWCNTs	Current density (1×10^{10} A/cm^2)	Highest temperature (K)	Equivalent resistance (KΩ)	Comments breakdown resistance
10	5	14	4.0	360	26.8	No
	10	52	1.0	303	7	No
23	5	14	9.2	820	27	Critical

6.5.6 Performances in terms of S-parameters

Based on performance, different CNT-based interconnects can be compared. Most importantly, resistance, inductance and capacitance of interconnect and high frequency response can be taken as performance parameters. Other performance matrices can be derived based on these basic parameters. The interconnect can be modeled as two port network as described in Section 6.4.3. Two ports network S_{11} and S_{21} parameters help understand the frequency response of an interconnect in terms of back scattering power and transmitted power. S_{11} is the ratio of power reflected from the

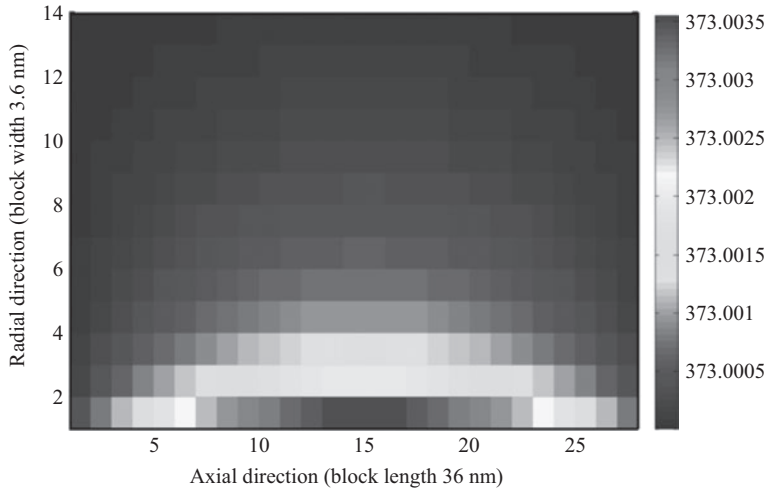


Figure 6.10 Temperature profile of MWCNT interconnect cross-section

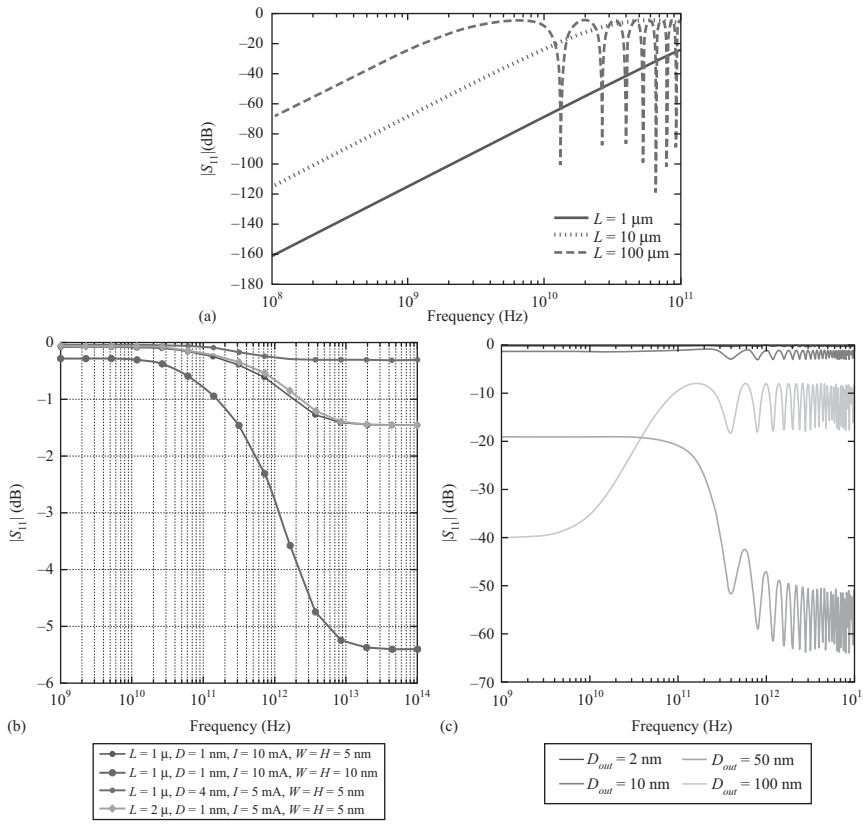

(a)
$|S_{11}|$(dB)
Frequency (Hz)
L = 1 µm
L = 10 µm
L = 100 µm
(b)
$|S_{11}|$ (dB)
Frequency (Hz)
L = 1 µ, D = 1 nm, I = 10 mA, W = H = 5 nm
L = 1 µ, D = 1 nm, I = 10 mA, W = H = 10 nm
L = 1 µ, D = 4 nm, I = 5 mA, W = H = 5 nm
L = 2 µ, D = 1 nm, I = 5 mA, W = H = 5 nm
(c)
$|S_{11}|$ (dB)
Frequency (Hz)
D_{out} = 2 nm
D_{out} = 50 nm
D_{out} = 10 nm
D_{out} = 100 nm

Figure 6.11 Scattering parameter S_{11} of various interconnects (a) SWCNT, (b) SWCNT bundle based and (c) MWCNT

transmission line to the incident power, while S_{21} is the ratio of power transmitted through the transmission line to the incident power.

$$S_{11} = 10\log\frac{P_{back}}{P_{incident}}; \quad S_{21} = 10\log\frac{P_{transmitted}}{P_{incident}} \tag{6.31}$$

For CNT interconnect, *S*-parameter calculations have been carried out with distributed elements and normalized by intrinsic impedance 50 Ω. Method described in Reference 51 can be used for *S*-parameter calculation per unit values of resistances, inductances and capacitances described in Section 6.4. In Figure 6.11, few of the results are shown regarding various interconnects covered so far in this chapter. From Figure 6.11, it is apparent that back scattering is high in high frequency range for SWCNT-bundle- and MWCNT-based interconnects. In case of SWCNT, back scattering is low in high frequency range. SWCNT could be of a potential use in high frequency circuits. In MWCNT interconnect higher outer diameter suffers more back scattering than the lower one. From Figure 6.12, it can be concluded that in general power transmission is high at high frequencies in these three kinds of CNT-based interconnects. Joule heating worsens the situation by reducing the transmission.

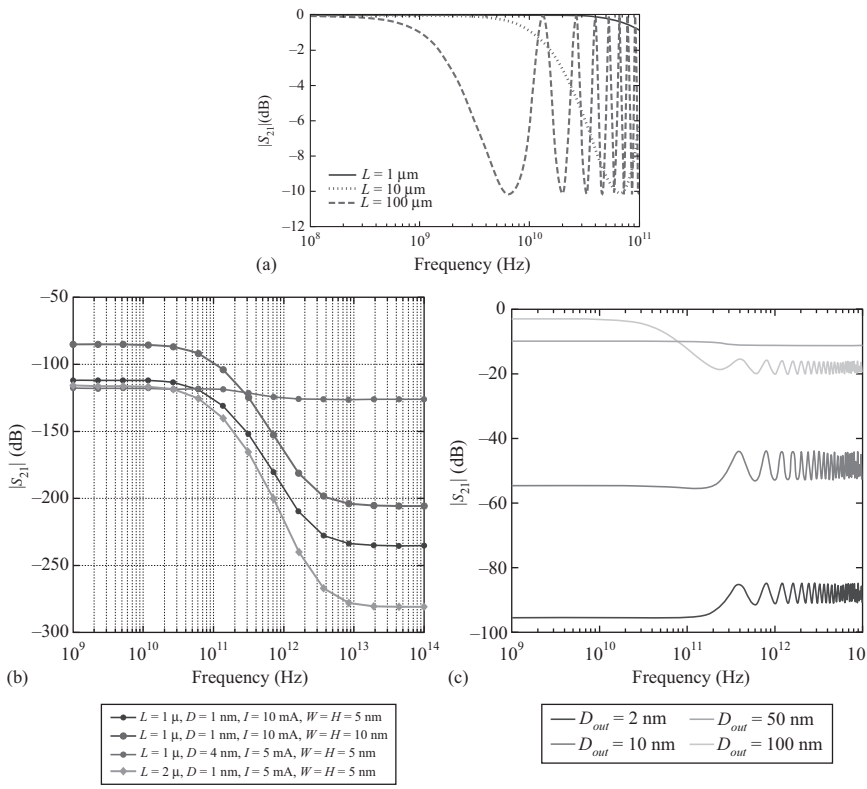

|S21|(dB)
L = 1 μm
L = 10 μm
L = 100 μm
Frequency (Hz)
(a)
|S21| (dB)
Frequency (Hz)
(b)
|S21| (dB)
Frequency (Hz)
(c)
L = 1 μ, D = 1 nm, I = 10 mA, W = H = 5 nm
L = 1 μ, D = 1 nm, I = 10 mA, W = H = 10 nm
L = 1 μ, D = 4 nm, I = 5 mA, W = H = 5 nm
L = 2 μ, D = 1 nm, I = 5 mA, W = H = 5 nm
Dout = 2 nm
Dout = 50 nm
Dout = 10 nm
Dout = 100 nm

Figure 6.12 Scattering parameter S_{12} of various interconnects (a) SWCNT, (b) SWCNT bundle based and (c) MWCNT

Depending on application a particular type of interconnect material can be preferred over the other one.

6.6 Conclusion

In this chapter, limitations of present CMOS VLSI interconnect technology has been addressed. Theory of CNT-based VLSI interconnects has been presented from their electrical properties and thermal properties. How to couple electrical and thermal equations has been also discussed and results have been presented. Although not the most accurate, to understand VLSI circuits, these models can be helpful in quick estimation of circuit performances. However, an accurate estimation of electro-thermal properties requires computationally highly complex first principle based quantum molecular dynamics, which is beyond the scope of this chapter. It is to be noted that significant research is underway to understand the complexity of phonon transport at the atomistic level. It is also to be emphasized that phonon transport is vital in understanding the Joule heating and how it affects the electron transport in low-dimensional VLSI interconnect. Phenomenological Fourier heat equation, which is known as heat diffusion equation, does not estimate the experimental results at the nanoscale. The reason is that if phonons mean free path is longer than the interconnect length then transport is no more diffusive, rather it is ballistic. This is why this heat diffusion equation is under scrutiny these days. Phonon BTE may help to estimate the thermal conductivity in ballistic heat transport. However, phonon BTE has its own limitation for assuming phonon as the classical particle instead of a quantum one. Interfacial thermal and electrical conductivities to be measured or estimated for various materials of interest need to be considered in order to find high electro-thermal conductive materials.

References

[1] R. Changsup, K. Kee-Won, A. L. S. Loke, *et al*., "Microstructure and reliability of copper interconnects," *IEEE Transactions on Electron Devices,* vol. 46, no. 6, pp. 1113–1120, 1999.

[2] K. Kyung-Hoae, P. Kapur, and K. C. Saraswat, "Compact performance models and comparisons for gigascale on-chip global interconnect technologies," *IEEE Transactions on Electron Devices,* vol. 56, no. 9, pp. 1787–1798, 2009.

[3] S. Iijima, "Helical microtubules of graphitic carbon," *Nature,* vol. 354, no. 6348, pp. 56–58, 1991.

[4] N. Srivastava and K. Banerjee, "Performance analysis of carbon nanotube interconnects for VLSI applications," *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 383–390, 2005.

[5] S. Berber, Y.-K. Kwon, and D. Tománek, "Unusually high thermal conductivity of carbon nanotubes," *Physical Review Letters,* vol. 84, no. 20, pp. 4613–4616, 2000.

[6] A. Naeemi, R. Sarvari, and J. D. Meindl, "Performance comparison between carbon nanotube and copper interconnect for gigascale integration (GSI)," *IEEE Electron Device Letters,* vol. 26, no. 2, pp. 84–86, 2005.

[7] A. Naeemi and J. D. Meindl, "Carbon nanotube interconnects," *Annual Review of Materials Research,* vol. 39, no. 1, pp. 255–275, 2009.

[8] A. Srivastava, Y. Xu, and A. K. Sharma, "Carbon nanotubes for next generation very large scale integration interconnects," *Journal of Nanophotonics,* vol. 4, no. 1, pp. 041690, 2010.

[9] E. Pop, D. A. Mann, K. E. Goodson, and H. Dai, "Electrical and thermal transport in metallic single-wall carbon nanotubes on insulating substrates," *Journal of Applied Physics,* vol. 101, no. 9, pp. 093710, 2007.

[10] J. Hone, M. Whitney, C. Piskoti, and A. Zettl, "Thermal conductivity of single-walled carbon nanotubes," *Physical Review B,* vol. 59, no. 4, pp. R2514–R2516, 1999.

[11] P. Kim, L. Shi, A. Majumdar, and P. L. McEuen, "Thermal transport measurements of individual multiwalled nanotubes," *Physical Review Letters,* vol. 87, no. 21, pp. 215502, 2001.

[12] L. Hong, X. Chuan, and K. Banerjee, "Carbon nanomaterials: the ideal interconnect technology for next-generation ICs," *IEEE Design & Test of Computers,* vol. 27, no. 4, pp. 20–31, 2010.

[13] M. Nihei, A. Kawabata, D. Kondo, *et al.*, "Electrical properties of carbon nanotube bundles for future via interconnects," *Japanese Journal of Applied Physics,* vol. 44, no. 4A, pp. 1626–1629, 2005.

[14] K. M. Mohsin, Y. M. Banadaki, and A. Srivastava, "Metallic single-walled carbon nanotube based temperature sensor with self heating," *Proceedings of SPIE (Smart Structures/NDE: Nano-Bio-, and Info-Tech Sensors and Systems: SSNO6)*, pp. 906003-906003-7, 2014.

[15] Y. M. Banadaki, K. M. Mohsin, and A. Srivastava, "A graphene field effect transistor for high temperature sensing applications," *Proceedings of SPIE (Smart Structures/NDE: Nano-Bio-, and Info-Tech Sensors and Systems: SSNO6)*, pp. 90600F-90600F-7, 2014.

[16] D. Jie and H. S. P. Wong, "A compact SPICE model for carbon-nanotube field-effect transistors including nonidealities and its application; Part I: model of the Intrinsic Channel Region," *IEEE Transactions on Electron Devices,* vol. 54, no. 12, pp. 3186–3194, 2007.

[17] S. Sinha, A. Balijepalli, and C. Yu, "Compact model of carbon nanotube transistor and interconnect," *IEEE Transactions on Electron Devices,* vol. 56, no. 10, pp. 2232–2242, 2009.

[18] Y. M. Banadaki, and A. Srivastava, "A novel graphene nanoribbon field effect transistor for integrated circuit design," *Proc. 56th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 924–927, 2013.

[19] A. Srivastava, Y. M. Banadaki, and M. S. Fahad, "(Invited) Dielectrics for Graphene Transistors for Emerging Integrated Circuits," ECS Transactions, Chapter 9, vol. 61, pp. 351–361, 2014.

[20] K. Sun, M. A. Stroscio, and M. Dutta, "Thermal conductivity of carbon nanotubes," *Journal of Applied Physics,* vol. 105, no. 7, pp. 074316, 2009.

[21] X. Yao, A. Srivastava, and A. K. Sharma, "A model of multi-walled carbon nanotube interconnects," *Proc. 52nd IEEE International Midwest Symposium on Circuits and Systems*, pp. 987–990, 2009.

[22] M. S. Sarto, and A. Tamburrano, "Single-conductor transmission-line model of multiwall carbon nanotubes," *IEEE Transactions on Nanotechnology*, vol. 9, no. 1, pp. 82–92, 2010.

[23] M. D'Amore, M. S. Sarto, and A. Tamburrano, "SPICE-model of multi-wall carbon nanotube through-hole vias," *Proc. Asia-Pacific Symposium on Electromagnetic Compatibility (APEMC)*, pp. 1104–1107, 2010.

[24] C. Ting-Yen, B. Shieh, and K. C. Saraswat, "Impact of Joule heating on scaling of deep sub-micron Cu/low-k interconnects," *Proc. Digest of Technical Papers in Symposium on VLSI Technology*, pp. 38–39, 2002.

[25] T. Yamada, T. Saito, D. Fabris, and C. Y. Yang, "Electrothermal analysis of breakdown in carbon nanofiber interconnects," *IEEE Electron Device Letters,* vol. 30, no. 5, pp. 469–471, May, 2009.

[26] H. Kitsuki, T. Yamada, D. Fabris, *et al.*, "Length dependence of current-induced breakdown in carbon nanofiber interconnects," *Applied Physics Letters,* vol. 92, no. 17, pp. 173110–173112, 2008.

[27] L. Forró, and C. Schönenberger, *"Physical properties of multi-wall nanotubes," in Carbon Nanotubes: Synthesis, Structure, Properties and Applications*, M. S.Dresselhaus, G. Dresselhaus and P. Avouris, eds., Berlin: Springer, 2000.

[28] Y. L. Kim, B. Li, X. An, *et al.*, "Highly aligned scalable platinum-decorated single-wall carbon nanotube arrays for nanoscale electrical interconnects," *ACS Nano,* vol. 3, no. 9, pp. 2818–2826, 2009/09/22, 2009.

[29] K. M. Mohsin, A. Srivastava, A. K. Sharma, and C. Mayberry, "A thermal model for carbon nanotube interconnects," *Nanomaterials,* vol. 3, no. 2, pp. 229–241, 2013.

[30] C. Forestiere, A. Maffucci, and G. Miano, "On the evaluation of the number of conducting channels in multiwall carbon nanotubes," *IEEE Transactions on Nanotechnology,* vol. 10, no. 6, pp. 1221–1223, 2011.

[31] K. M. Mohsin, A. Srivastava, A. K. Sharma, and C. Mayberry, "Characterization of MWCNT VLSI interconnect with self-heating induced scatterings," *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, pp. 368–373, 2014.

[32] P. J. Burke, "An RF circuit model for carbon nanotubes," *IEEE Transactions on Nanotechnology,* vol. 2, no. 1, pp. 55–58, 2003.

[33] R. Stephan, J. Jie, E. F. F. T. Luis, and S. Riichiro, "Charge transport in carbon nanotubes: quantum effects of electron–phonon coupling," *Journal of Physics: Condensed Matter,* vol. 19, no. 18, p. 183203, 2007.

[34] M. D'Amore, M. S. Sarto, and A. Tamburrano, "Modelling of multiwall carbon nanotube transmission lines," *Proceedings of International Conference on Electromagnetics in Advanced Applications, 2007. ICEAA 2007*, pp. 629–632, 2007.

[35] M. S. Sarto and A. Tamburrano, "Multiconductor transmission line modeling of SWCNT bundles in common-mode excitation," *Proceedings of IEEE International Symposium on Electromagnetic Compatibility, 2006. EMC 2006.*, pp. 466–471, 2006.

[36] D. G. Cahill, W. K. Ford, K. E. Goodson, *et al*., "Nanoscale thermal transport," *Journal of Applied Physics,* vol. 93, no. 2, pp. 793–818, 2003.

[37] D. G. Cahill, P. V. Braun, G. Chen, *et al.*, "Nanoscale thermal transport. II. 2003–2012," *Applied Physics Reviews,* vol. 1, no. 1, p. 011305, 2014.

[38] C. A. Santini, P. M. Vereecken, A. Volodin, *et al.*, "A study of Joule heating-induced breakdown of carbon nanotube interconnects," *Nanotechnology,* vol. 22, no. 39, pp. 395202–395210, 2011.

[39] E. Pop, D. Mann, J. Reifenberg, K. Goodson, and H. Dai, "Electro-thermal transport in metallic single-wall carbon nanotubes for interconnect applications, " *Proceedings of IEDM Technical Digest*, pp. 456–459, 2005.

[40] N. Y. Huang, J. C. She, J. Chen, *et al.*, "Mechanism responsible for initiating carbon nanotube vacuum breakdown," *Physical Review Letters,* vol. 93, no. 7, pp. 075501–075504, 2004.

[41] E. Pop, D. Mann, Q. Wang, K. Goodson, and A. H. Dai, "Thermal conductance of an individual single-wall carbon nanotube above room temperature," *Nano Letters,* vol. 6, no. 1, pp. 96–100, 2006.

[42] S. Bhattacharya, R. Amalraj, and S. Mahapatra, "Physics-based thermal conductivity model for metallic single-walled carbon nanotube interconnects," *IEEE Electron Device Letters,* vol. 32, no. 2, pp. 203–205, 2011.

[43] P. Kim, L. Shi, A. Majumdar, and P. McEuen, "Thermal transport measurements of individual multiwalled nanotubes," *Physical Review Letters,* vol. 87, no. 21, p. 215502, 2001.

[44] A. E. Aliev, M. H. Lima, E. M. Silverman, and R. H. Baughman, "Thermal conductivity of multi-walled carbon nanotube sheets: radiation losses and quenching of phonon modes," *Nanotechnology,* vol. 21, no. 3, pp. 035709, 2010.

[45] H. Hayashi, T. Ikuta, T. Nishiyama, and K. Takahashi, "Enhanced anisotropic heat conduction in multi-walled carbon nanotubes," *Journal of Applied Physics,* vol. 113, no. 1, pp. 014301–014304, 2013.

[46] L. Feng, W. Gaofeng, and L. Hai, "Modelling of self-heating effects in multi-wall carbon nanotube interconnects," *Micro & Nano Letters, IET,* vol. 6, no. 1, pp. 52–54, 2011.

[47] L. Feng, W. Gaofeng, and L. Hai, "Modeling of crosstalk effects in multiwall carbon nanotube interconnects," *IEEE Transactions on Electromagnetic Compatibility*, vol. 54, no. 1, pp. 133–139, 2012.

[48] M. Nihei, D. Kondo, A. Kawabata, *et al.*, "Low-resistance multi-walled carbon nanotube vias with parallel channel conduction of inner shells [IC interconnect applications]," *Proceedings of the IEEE 2005 International Interconnect Technology Conference*, pp. 234–236, 2005.

[49] H. Li, W. Lu, J. Li, X. Bai, and C. Gu, "Multichannel ballistic transport in multiwall carbon nanotubes," *Physical Review Letters,* vol. 95, no. 8, pp. 086601–086604, 2005.

[50] K. M. Mohsin and A. Srivastava, "Characterization of SWCNT bundle based VLSI interconnect with self-heating induced scatterings," *Proceedings of* GLSVLSI '15, pp. 265–270, 2015.

[51] A. A. Bhatti, "A computer based method for computing the n-dimensional generalized abcd parameter matrices of n-dimensional systems with distributed parameters," *Proceedings of 22nd Southeastern Symposium on System Theory (SSST) Conference*, pp. 590–593, 1990.

Chapter 7

Lumped electro-thermal modeling and analysis of carbon nanotube interconnects

Aida Todri-Sanial[1]

Carbon nanotubes (CNTs) due to their unique electrical, thermal, and mechanical properties are being investigated as promising candidate material for on-chip and off-chip interconnects. The attractive mechanical properties of CNTs, including high Youngs modulus, resiliency, and low thermal expansion coefficient, offer great advantage for reliable and strong interconnects, and even more so for local and global on-chip interconnects. With aggressive scaling, on-chip interconnects contribute to power consumption and heat build-up due to their increasing parasitics with scaling which detriment overall energy efficiency of circuits. Due to their unique properties, CNTs present an opportunity to address these challenges and provide solutions for reliable signal and power/ground interconnects. In this chapter, we perform detailed electro-thermal analyses of horizontally aligned CNTs and report on their performance and voltage drop.

7.1 Introduction

CNTs are a class of nanomaterials with unique mechanical, thermal, and electrical properties [1]. CNTs can be classified into two types: single-wall (SWCNTs) and multi-wall (MWCNTs). SWCNTs are rolled graphitic sheets with diameters on the order of 1 nm. MWCNTs consist of several rolled graphitic sheets nested inside each other and can have diameters as large as 100 nm. Depending on their chirality, the CNTs can be metallic or semiconductors. Metallic CNTs (m-CNTs) are ballistic conductors, which show promise for use as interconnects in nanoelectronics. On the other hand, semiconducting CNTs (s-CNTs) have a diameter-depended band-gap and do not have surface states that need passivation, thus can be used to make devices such as diodes and transistors [1–4].

CNTs are cylindrical carbon molecules formed by one-atom thick sheets of carbon, or graphene. CNTs, both SWCNT and MWCNT, are being investigated for a

[1]CNRS-LIRMM, Montpellier, France

variety of nanoelectronics applications because of their unique properties [1]. Their extraordinary large electron mean free paths and resistance to electromigration make them potential candidates for interconnects in large-scale systems. During the past decade, most of research is focused on CNT growth, synthesis, modeling and simulation and characterizing contact interfaces [5, 6]. Detailed simulation for signal interconnects has been performed by References 2 and 3 and shown that CNTs have lower parasitics than Cu metal lines, however, the contact resistance between CNT-to-CNT and CNT-to-metal is large and can be detrimental for timing issues. Additionally, researchers are looking into different CNT growth techniques that are compatible with CMOS process and lab measurements indicate the potential of integrating CNTs on-chip [7, 8].

One essential and most interesting application of the nanotubes in microelectronics is as interconnects using the ballistic (without scattering) transport of electrons and the extremely high thermal conductivity along the tube axis [9]. Electronic transport in SWCNTs and MWCNTs can go over long nanotube lengths, 1 μm, enabling CNTs to carry very high currents (i.e. $>10^9$ A/cm^2) with essentially no heating due to nearly 1 D electronic structure.

In literature, the comparison of copper and CNTs has been limited to signal interconnects. Investigation of CNTs for power and clock delivery would also have a significant importance. It would reveal whether or not CNTs can potentially replace both signal and power/ground copper wires. Additionally, clock and power networks are most vulnerable to electromigration, it is therefore critical to know whether or not CNTs improve their reliability. Additionally, most of the CNT modeling and investigations are focused on their electrical properties, whereas few works exist that look into their electro-thermal modeling and properties.

From the large body of research related to CNT analysis, mainly two groups of works can be identified. The first group of works focuses on modeling aspects of CNT interconnects [1, 4, 9, 10]. The second group of works focuses on performance comparison of CNT interconnects versus copper (Cu) interconnects [2, 4, 5, 11]. Almost all these works have considered the application of CNT interconnects for signaling and few works focus on power delivery [3, 12]. Complementary to these efforts, in this chapter, we investigate electro-thermal properties of horizontally aligned CNTs for signal interconnects and power/ground delivery network.

The rest of this chapter is organized as follows. Section 7.2 provides a detailed description of electrical properties of CNT interconnects. Section 7.3 provides description and discussions on electro-thermal modeling of CNTs where we present some electro-thermal analysis for CNTs as signal interconnects and power/ground delivery networks. Section 7.4 concludes this chapter.

7.2 Electrical modeling of CNTs

There are many papers in literature that focus on CNT modeling and understanding its transport properties [1–4, 9]. In this section, we provide a brief description of CNT modeling that we utilize in this work. A generalized model for CNT interconnects

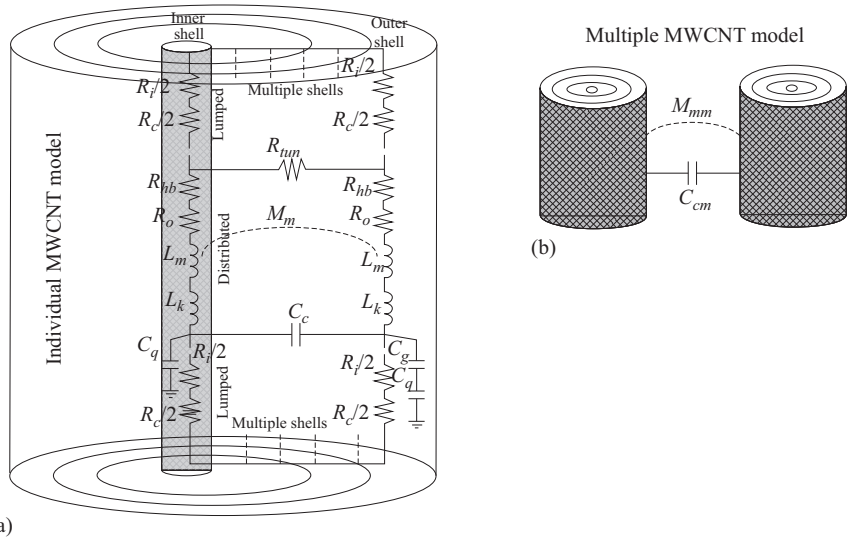


Figure 7.1 (a) Circuit model of an individual MWCNT and (b) multiple MWCNTs. This is general enough model to be applicable to MWCNTs of different diameters and shell numbers. It can also be applicable to SWCNTs where the model of a single shell can be utilized

is depicted as in Figure 7.1. In Figure 7.1a, the model of an individual MWCNT is shown with parasitics represent both dc conductance and high-frequency impedance i.e. inductance and capacitance effects. Multiple shells of a MWCNT are presented by the individual parasitics of each shell. Such model can also be applicable to SWCNTs where only a single shell is represented.

Each shell has a lumped ballistic resistance (R_i) and lumped contact resistance (R_c) due to imperfect metal–nanotube contacts. These contacts are typically constructed of gold, palladium, or rhodium [1]. The nanotubes have also a distributed ohmic resistance (R_o), which is dependent on length, l_b, and mean free path of acoustic phonon scattering (λ_{ap}). Overall CNT resistance depends also on the applied bias voltage, $R_{hb} = V_{bias}/I_o$, where I_o is the maximum saturation current (I_o values 15–30 μA [10]) Between shells in MWCNTs, there is also an intershell tunneling resistance (R_{tun}). As the applied bias voltage to each shell is the same, the impact of R_{tun} is relatively small. All the aforementioned ballistic, ohmic, and contact resistances depend on the number of 1D conducting channels, N_c. For metallic SWCNTs, the number of conducting channels is always $N_c = 2$ due to lattice degeneracy [9]. Whereas, for semiconducting SWCNTs and small diameter semiconducting MWCNTs, $N_c = 0$. For any conducting shells in an MWCNT, the intrinsic resistance, $R_i = R_q/N_c$, where R_q is the quanta conductance for a 1D conduction channel ($R_q = 12.9\,\Omega$) [9]. Also, contact resistance is $R_c = 2R_{co}/N_c$ where R_{co} is the nominal contact resistance [9]. Ohmic resistance is derived as $R_o = R_q L/N_c d_s C_\lambda$, where L is the length of the

MWCNT, d_s is the diameter of the shell, and C_λ is the acoustic phonon scattering mean free path (λ_{ap}). Thus, the total resistance of an individual nanotube (R_t) can be obtained by computing resistance of each shell, $R_{shell} = R_i + R_c + R_{hb} + R_o$:

$$G_t = \sum_{i=0}^{N} \frac{1}{R_{shell_i}} = \sum_{i=0}^{N} \frac{1}{\frac{R_f}{N_c} + \frac{R_q L}{C_\lambda (d_{in} + iS_a) N_c}} \tag{7.1}$$

where $R_f = R_i + R_c + R_{hb}$, d_{in} is the diameter of the inner shell, S_a is the space between shells where typically 0.34 nm is shell thickness, and 0.34 nm is shell-to-shell spacing, and N is the number of shells in MWCNT as $N = (d_{out} - d_{in})/S_a$ where d_{out} is the diameter of the outer shell. In a bundle of SWCNTs or MWCNTs, the total resistance can be derived as, $R_b = R_t/n_b$, where n_b is the number of bundles. For example, a metal track with width, w, and heigh, h, the number of horizontally aligned nanotube bundles can be expressed as in Reference 10:

$$n_b = P_m(n_h n_w - \lfloor n_h/2 \rfloor) \tag{7.2}$$

where P_m is the probability that a nanotube is metallic and usually $P_m = 0.3$ [9], n_h is the number of nanotubes in vertical direction as $n_h = \lfloor h/d_{out} \rfloor$, and n_w is the number of nanotubes in horizontal direction as $n_w = \lfloor w/d_{out} \rfloor$.

The capacitance of nanotubes consists of both quantum, C_q, and electrostatic capacitances, C_e, that can impact power supply noise on power tracks. Additionally, there is coupling capacitance between: (1) conducting shells in an individual MWCNT, C_c, and (2) individual MWCNTs depending on the proximity between them, C_{cm}. Using Luttinger liquid theory [9], quantum capacitance can be derived as $4e^2/h_p v_F \approx 193a\text{F}/\mu\text{m}$ per conducting channel where h_p is Planck's constant, e is charge of single electron, and v_F is Fermi velocity in graphene. Therefore, for each shell, quantum capacitance is as $C_q = \frac{4e^2}{h_p v_F} N_c L$, and the total quantum capacitance of a CNT bundle is as:

$$C_{q_t} = n_b \sum_{i=1}^{N} C_{q_i} \tag{7.3}$$

Electrostatic coupling depends on the geometry of the CNT and also the bundle density (i.e. number of bundles, n_b). It is shown in Reference 5 that CNT bundles have slightly smaller electrostatic capacitance compared to Cu interconnects with same dimensions. Capacitance of CNT bundles would decrease slowly with increase of bundle density [9]. However, for a MWCNT, these capacitances cannot be assumed equal due to the fringing coupling effects between shells. The electrostatic capacitance of a MWCNT which is equivalent to ground capacitance from the outer shell to the ground plane, distance y, can be obtained as:

$$C_{e_t} = \frac{2\pi\varepsilon}{ln(y/d_{out})} \tag{7.4}$$

The shell-to-shell coupling capacitance is as in [2, 5]:

$$C_c = \frac{2\pi\varepsilon}{ln(d_{out}/d_{in})} \tag{7.5}$$

and coupling capacitance C_{cm} between two CNT bundles with space, s can be expressed as:

$$C_{cm} = \frac{2\pi\varepsilon}{s/d_{out}} \tag{7.6}$$

As for inductance, CNTs have both kinetic and magnetic inductance that impact power supply noise and high-frequency effects on power tracks. Again, based on the Luttinger liquid theory, the kinetic inductance per conducting shell can be theoretically expressed as $L_k = h_p L/4e^2 v_F N_c$ or $\approx 8\,\text{nH}/\mu\text{m}$ per conducting shell. Thus, the total kinetic inductance for all shells in a CNT bundle is derived as:

$$L_{k_t} = \frac{1}{n_b \sum_{i=1}^{N} \frac{1}{L_{k_i}}} \tag{7.7}$$

where L_{k_i} is the kinetic inductance of each shell i. Magnetic inductance, L_m and mutual inductance M_m are also of importance as they can have an impact on dynamic voltage drop behavior. For each shell $L_m = \frac{\mu}{2\pi} ln(y/d)$ and for a CNT bundle is derived as:

$$L_{m_t} = \frac{1}{n_b \sum_{i=1}^{N} \frac{1}{L_{m_i}}} \tag{7.8}$$

Scalable mutual inductance model between any two shells i and $i+1$ with space distance, S_a was presented in [1, 4, 5] and can be estimated as:

$$M_{m_i} = \frac{\mu_o l}{\pi} ln(S_a/(d_{i+1} - d_i)) \tag{7.9}$$

and mutual inductance M_{mm}, between two CNT bundles with space, s can be similarly expressed as:

$$M_{mm} = \frac{\mu_o l}{\pi} ln(s/d_{out}) \tag{7.10}$$

Resistance, capacitance and inductance models for MWCNTs are further utilized to study the dynamic voltage drop and performance latencies for power delivery and signaling interconnects.

7.3 Thermal modeling for CNTs

In an integrated chip, the temperature may rise above $80°C$ which will impact the behavior of devices and parasitics of interconnects. As interconnect lengths are usually

larger than the free mean path of CNTs, there will be electron–phonon scattering along the length of nanotube that would lead to self-heating and temperature rise along the nanotube interconnect. The temperature variations along a nanotube are also dependent on the defect density, alignment, and contact resistance. Large contact resistances create large potential barriers at the interfaces of nanotubes for electrons to tunnel through. Additionally, large defect densities cause electrons to localize and conduction happens through thermally activated electron hopping. As already mentioned on the previous sections, there are many contradicting and inconsistent reports on the thermal conductivity and temperature coefficient of resistance (TCR) for CNTs. TCR is the change in resistance for every 1 K of temperature rise. In this section, we will exploit the existing physical models for CNTs and express its properties as a function of temperature.

We investigate both SWCNTs and MWCNTs with various lengths and diameters. We start by deriving the thermal coefficient for resistance for SWCNTs as in [13]:

$$TCR_{swcnt} = \frac{(L/10^3 d_s)/T_o}{1 + (L/10^3 d_s)(T/T_o - 2)} \tag{7.11}$$

where L and d_s are the length and shell diameter of the nanotube, respectively. T is the temperature and $T_o = 300$ K. For a single wall nanotube with small diameter, the number of conduction channels N_c is independent of temperature for long length interconnects (i.e. significantly larger than mean free path at room temperature). Whereas, for large diameter nanotubes with increasing temperature, the number of conduction channels increase which consequently increases N_c. Similarly, nanotube conductance also depends on nanotube length and temperature. Neutral length L_N is the length at which resistance becomes independent of temperature. For lengths smaller than L_N, the nanotube resistance is mainly influenced by the number of conduction channels, and increasing temperature lowers resistance. Whereas, for lengths larger than L_N, the mean free path is more important, and increasing temperature increases resistance. Neutral length is derived as in [13]:

$$L_N = \frac{10^3 a T_o d_s^2}{b + 2a T_o d_s} \tag{7.12}$$

where a is 2.04×10^{-4} nm^{-1}K^{-1} and $b = 0.425$. Figure 7.2 shows the thermal coefficient of resistance for SWCNTs with different lengths and diameters. We observe that small diameter nanotubes have larger TCR than nanotubes with large diameter. For example, for nanotubes with length 1 μm and diameter 1 nm, the TCR = 6.5 whereas for nanotubes with diameter 10 nm, the TCR = 4. For short length and large diameter nanotubes, the TCR is relative small (i.e. <1), then TCR increases linearly with nanotube length. It is important to note that TCR is a positive coefficient for SWCNTs with different lengths and diameters.

In Figure 7.3, the ratio of resistances with respect to resistance at $T = 300$ K is shown for nanotubes with diameter 1 nm and various lengths. We note that for short

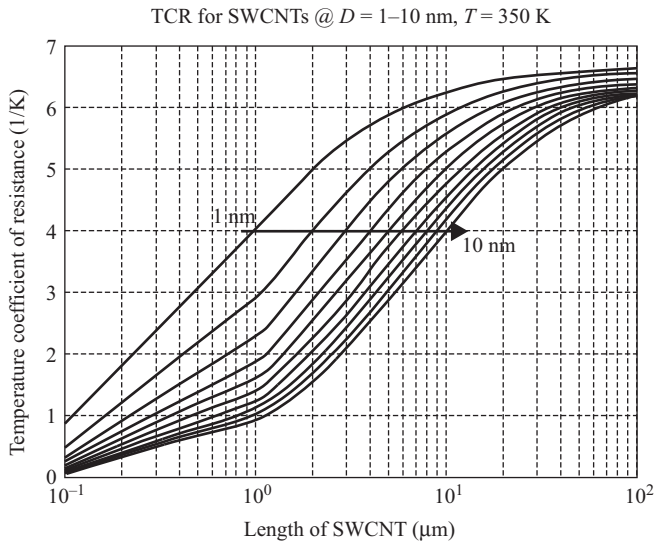


Figure 7.2 TCR for SWCNTs

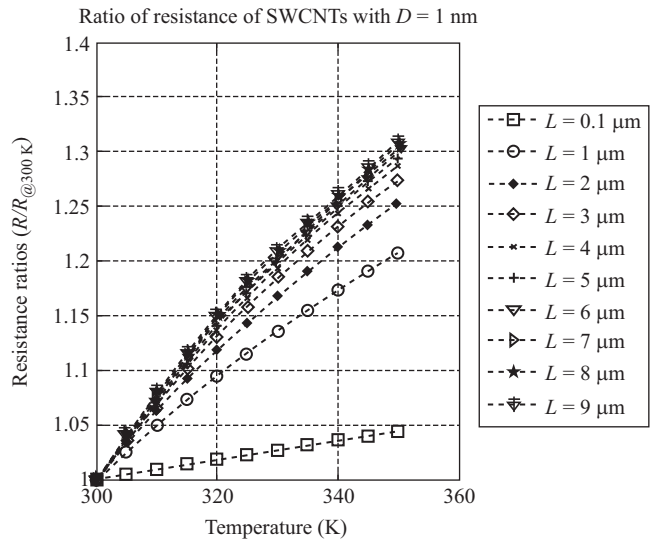


Figure 7.3 Ratio of resistances due to temperature impact for SWCNTs with d = 1 nm

length nanotubes, the resistance ratio is small (i.e. <1.05) for temperature ranges of 300–350 K. There is a linear increase to the resistance ratio as the nanotube length increases (i.e. ratio = 1.3 for $L = 9\,\mu m$). Hence, the nanotube resistance increases with temperature but at different rates depending on nanotube length. A similar plot

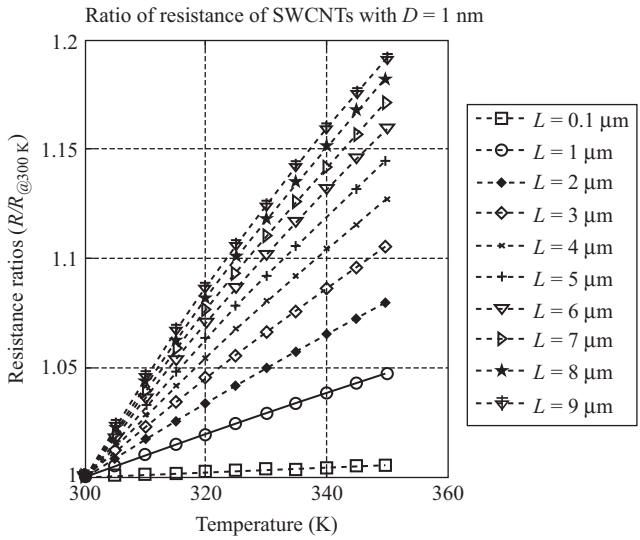


Figure 7.4 Ratio of resistances due to temperature impact for SWCNTs with $d = 10\,nm$

is shown in Figure 7.4 where the nanotube diameter is set to 10 nm. We observe a similar trend as in Figure 7.3, but the resistance ratios are smaller. This is an important observation for nanotube interconnects where diameter and length of nanotubes can be exploited as a knob for alleviating temperature impact on resistance. For example, the resistance ratio is minor when increasing nanotube diameter from 1 nm to 10 nm for nanotube lengths of 10 μm.

In Figure 7.5, the resistance distribution is shown for SWCNTs with different lengths and temperature values for nanotubes of $d = 1$ nm. As already mentioned above, for short length nanotubes, the temperature increase does not change the conduction channels, hence no change in resistance. Whereas, as the length increases, it also increases conduction channels, N_c, which further increase resistance. Additionally, as temperature increases, we observe an increase on resistance due to temperature impact on N_c.

To derive TCR for MWCNTs, it is important to understand the heat transport and distribution on individual shells and bundles. Once heat is introduced in the outer shell of MWCNT, the high thermal conductivity along the graphene layer transfer heat at a high flow rate in the circumferential direction, as well as along the tube. Due to close proximity between shells, there is thermal coupling that enables heat flow between shells. Authors in Reference 14 demonstrated that heat introduced at outer shells is evenly distributed to all shells within a short distance, $L \sim 50$ nm. The total heat flowing through the outer shell is always higher than the current in the inner shells. We compute the thermal coefficient of resistance based for each shell, which then can be used to compute the resistance of MWCNT with respect

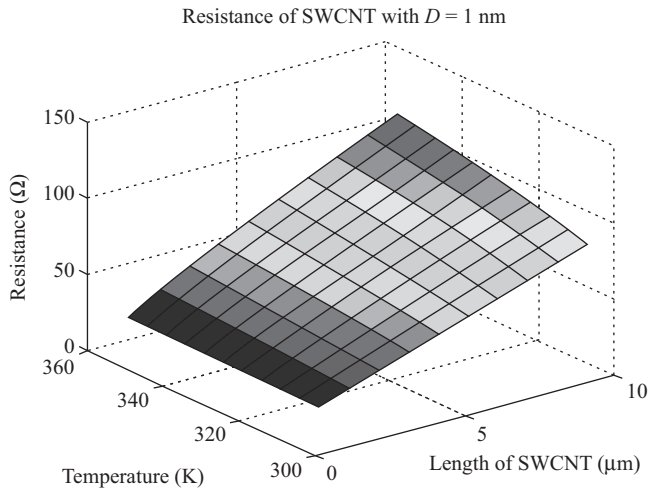


Figure 7.5 Resistance distribution for SWCNTs with diameter = 1 nm for various lengths and temperature values

to temperature. As the outer shell conducts most of the heat, we derive its TCR as in [13]:

$$TCR_{mwcnt} = \frac{\delta R/\delta T}{R_{i_{th}}} \tag{7.13}$$

where

$$\frac{\delta R}{\delta T} = \left(\frac{R_q}{10^3 d_{out}}\right)\left(\frac{2ad_{out} + b/T_o}{(aTd_{out} + b)^2}\right)(L - L_N) \tag{7.14}$$

$$R_{i_{th}} = R_q + \left(\frac{R_q L}{\lambda_{ac}} + \frac{R_q L}{\lambda_{op\_abs}} + \frac{R_q L}{\lambda^{fld}_{op\_ems}} + \frac{R_q L}{\lambda^{abs}_{op\_ems}}\right)/N_c \tag{7.15}$$

where the mean free path for acoustic scattering is as [15] $\lambda_{ac} = 10^3 d_{out} T_1/T$ where $T_1 = 400\,\text{K}$. Optical phonon scattering can occur if an electron obtains adequate energy (i.e. $\hbar w_{op} \approx 0.18\,\text{eV}$), it can emit an optical phonon and get backscattered. The scattering length is (much shorter than acoustic scattering) computed as $\lambda_{op} = 56 d_{out}$ and measured with smaller coefficients (i.e. ~15–20) [16]. The mean free path for absorbing an optical phonon is as $\lambda_{op\_abs} \approx \lambda_{op}/n_{phonons}$ where $n_{phonons} = 1/(e^{\hbar w/K_B T} - 1)$ is the number of phonons and K_B Boltzmann constant.

An electron can obtain sufficient energy for emitting an optical phonon either by getting accelerated long enough by electrical field or by absorbing an optical phonon. The scattering lengths can be calculated as:

$$\lambda^{fld}_{op\_ems} = \hbar w_{op}/(qV_{bias}/L) + \lambda_{op} \tag{7.16}$$

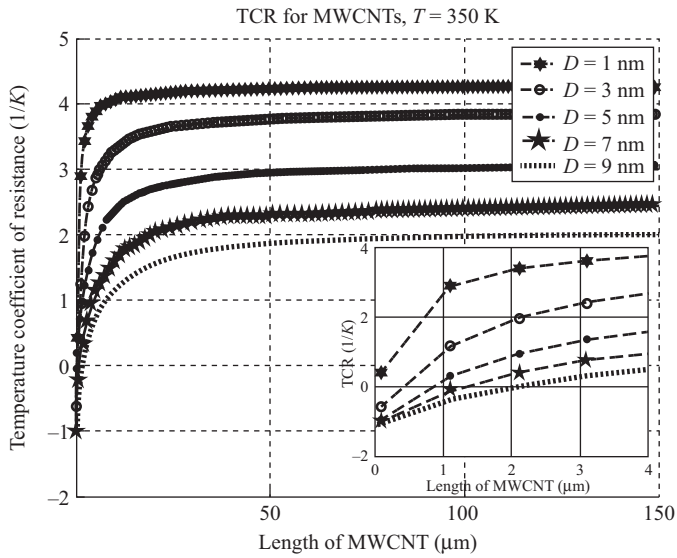

TCR for MWCNTs, T = 350 K
D = 1 nm
D = 3 nm
D = 5 nm
D = 7 nm
D = 9 nm
Temperature coefficient of resistance (1/K)
Length of MWCNT (μm)
TCR (1/K)
Length of MWCNT (μm)

Figure 7.6 TCR for MWCNTs for longer lengths (i.e. up to 150 μm) and outer diameters (i.e. 1–10 nm). The inset shows the TCR for shorter lengths MWCNTs

and

$$\lambda_{op\_ems}^{abs} = \lambda_{op\_abs} + \lambda_{op} \tag{7.17}$$

where V_{bias} is the applied voltage. It has been shown that increasing the diameter or temperature linearly increases the number of conduction channels in large shells [13]. The average number of channels in a shell can be estimated as :

$$N_c \approx \begin{cases} aTd_s + b, & d_{out} > d_T/T \\ 2/3, & d_{out} < d_T/T \end{cases} \tag{7.18}$$

where d_T is 1300 nm K, whose value is determined by the thermal energy of electrons.

Figure 7.6 shows the TCR for MWCNTs for various lengths and diameters. The inset figure shows the TCR for shorter length MWCNTs. It is important to note that the TCR for large diameter (i.e. $d_{out} > 1$ nm) is negative, where the increase in temperature leads to decrease in resistance. This behavior can be explained by Joule heating (or self-heating) effect and is consistent with negative TCRs obtained by References 17 and 13. Theoretical analysis has also shown that negative TCRs for MWCNTs [13, 18]. Negative TCRs can be explained from the fact that there are more channels in MWCNTs contributing to conductance at higher temperatures as per Fermi–Dirac distribution. Larger number of channels lowers both scattering resistance and contact resistance. The negative TCR is opposing with other metals

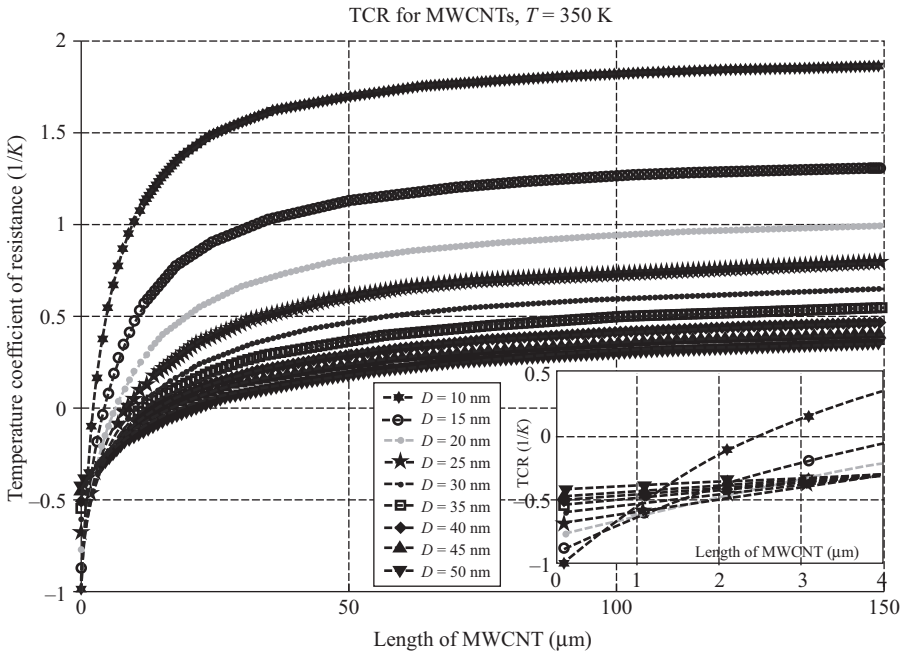

TCR for MWCNTs, T = 350 K
Temperature coefficient of resistance (1/K)
Length of MWCNT (μm)
D = 10 nm
D = 15 nm
D = 20 nm
D = 25 nm
D = 30 nm
D = 35 nm
D = 40 nm
D = 45 nm
D = 50 nm
TCR (1/K)
Length of MWCNT (μm)

Figure 7.7 TCR for MWCNTs for longer lengths (i.e. up to 150 μm) and outer diameters (i.e. 10–50 nm). The inset shows the TCR for smaller lengths MWCNTs

(i.e. copper) conductors, which presents an advantage for CNTs to be implemented as on-chip interconnect material. We also note that as length of nanotube increases, the TCR also increases till it saturates for lengths longer than 50 μm.

In Figure 7.7, the TCRs for large diameter MWCNTs are shown. The diameter varies from 10 nm to 50 nm for various nanotube lengths. The inset figure shows the TCR for short length MWCNTs. We note a negative TCR that for large-diameter and short-length MWCNTs. As length increases, TCR also increases and saturates for MWCNT lengths of 50 μm. In comparison with Figure 7.6, large diameter nanotubes have negative TCRs, and small increase in TCR is observed for longer length nanotubes.

From these experiments, we derive that temperature effects can be coped with for interconnects with MWCNTs where for short lengths even a decrease in resistance can be obtained. Figures 7.8 and 7.9 show the resistance distribution for MWCNTs with diameters 10 nm and 50 nm, respectively. The resistance of MWCNTs with diameter 10 nm is at least 1 × order of magnitude larger from nanotubes with diameter $d = 50$ nm for $T = 350$ K. Such observation is significant for application of CNTs as signaling and power interconnects where large timing errors and voltage drops would be attained. Such large changes in resistance at high temperatures would impact the dimensions of MWCNTs that can be used for building reliable interconnects.

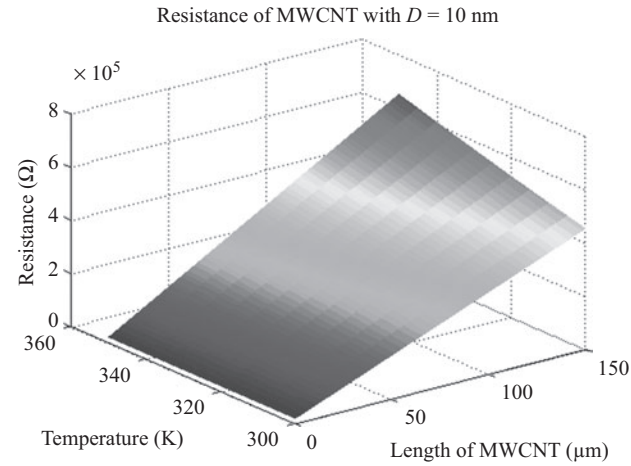


Figure 7.8 Resistance distribution for MWCNTs with diameter = 10 nm for various lengths and temperature values

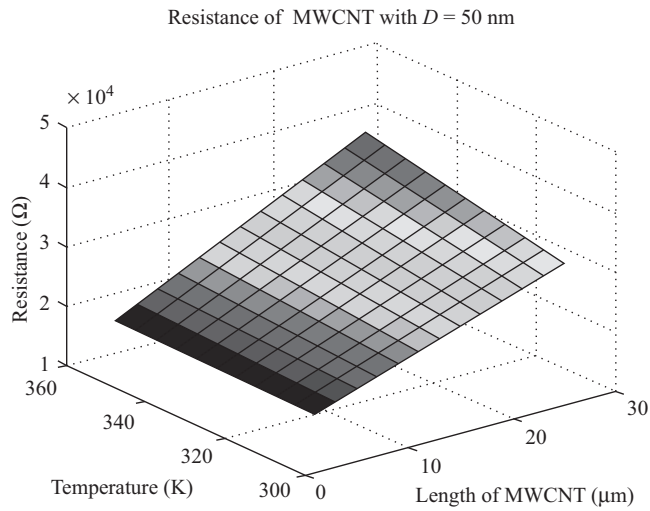


Figure 7.9 Resistance distribution for MWCNTs with diameter = 50 nm for various lengths and temperature values

Another important metric is the resistance ratio with respect to resistance at $T = 300$ K. Figures 7.10 and 7.11 show the resistance ratios for MWCNTs with diameters 10 nm and 50 nm, respectively. Depending on the diameter and temperature, the neutral length, L_N is obtained and also shown the title of each figure. L_N represents the MWCNT length that is independent of temperature. For MWCNT with diameter $d = 10$ nm, the neutral length is $L_N = 2.449\,\mu$m. This can also be deduced from Figure 7.10, where resistance ratio is less than 1 for MWCNT lengths of 1 μm.

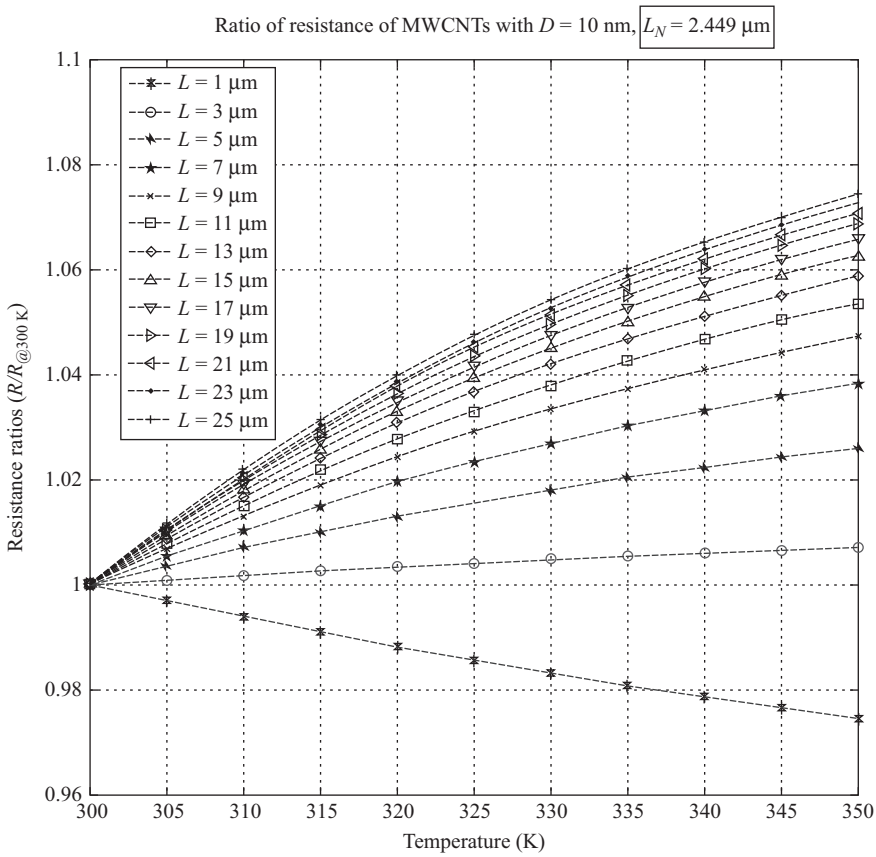


Figure 7.10 Resistance ratio for MWCNTs with $D = 10$ nm for various lengths. Note that $L_N = 2.449\ \mu m$

For MWCNT lengths of 3 μm, the resistance ratios are almost constant to 1 (i.e. as it is close to L_N). The resistance ratio linearly increases with increase in MWCNT length and reaching maximum of 7% increase (i.e. maximum resistance ratio 1.07 for 25 μm nanotube length).

Figure 7.11 shows resistance ratios for 50 nm diameter nanotubes where $L_N = 20.6\ \mu m$. Such large L_N means that long MWCNT interconnects can be used without any dominant impact from temperature. This can also be noted from Figure 7.11 where most of the resistance ratios are below 1. For example, MWCNTs with lengths $L < L_N$ have lower resistance ratio, and temperature has a negative impact on the resistance. Such observations are important for selecting interconnect dimensions that immune to joule heating effects.

To predict the voltage drop on a power delivery network, we analyze a single branch implemented with MWCNTs. To compute voltage drop on the branch, we make assumptions that current flowing on the branch, $I_{branch} = 1\ \mu A$, $dI_{branch} = 1\ \mu A$,

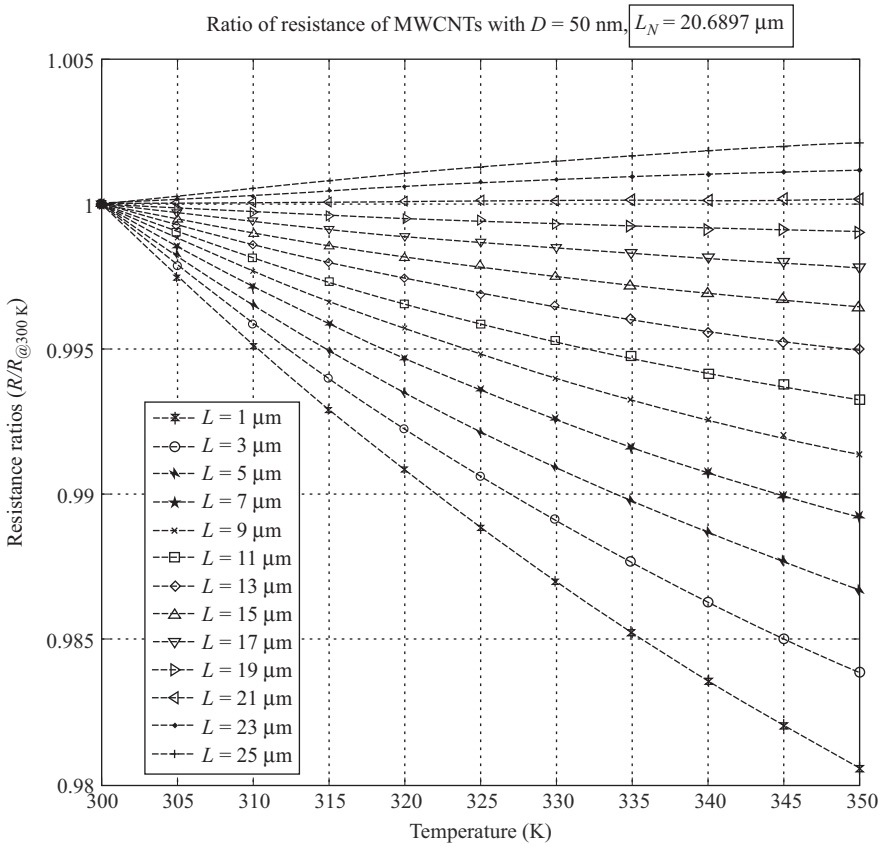


Figure 7.11 Resistance ratio for MWCNTs with $D = 50\,nm$ for various lengths. Note that $L_N = 20.6897\,\mu m$

$dt = 1$ ns (or 1 GHz switching frequency), and $dV_{branch} = 0.1$ V. Note that these values are simply chosen to quantify branch voltage drop when 1 µA current is flowing on the branch for varying MWCNT lengths and diameters. Figures 7.12 and 7.13 show the voltage drop for $T = 300$ K and $T = 350$ K, respectively. Overall, we observe minor voltage drop changes due to the impact of negative TCR on resistance. We deduce that selecting MWCNT interconnects length and diameter are essential for limiting the amount of voltage drop. We also compute the delay of MWCNT nanotube interconnects when it is driven by a gate with driver resistance of 200Ω, driver and load capacitances of 0.5 fF and 1 fF, respectively. The Elmore delay of the segment can be calculated as:

$$t_d = 0.69[R_{driver}C_{driver} + C_q(R_{driver} + R_{i_{th}}) + C_{load}(R_{driver} + R_{i_{th}} + R_c)] \quad (7.19)$$

where R_c is the contact resistance. Figures 7.14 and 7.15 show the delay for MWCNTs with different lengths and temperatures for diameters $d = 10$ nm and $d = 50$ nm, respectively. Overall, it is observed a delay increase up to 5.5% for long-length

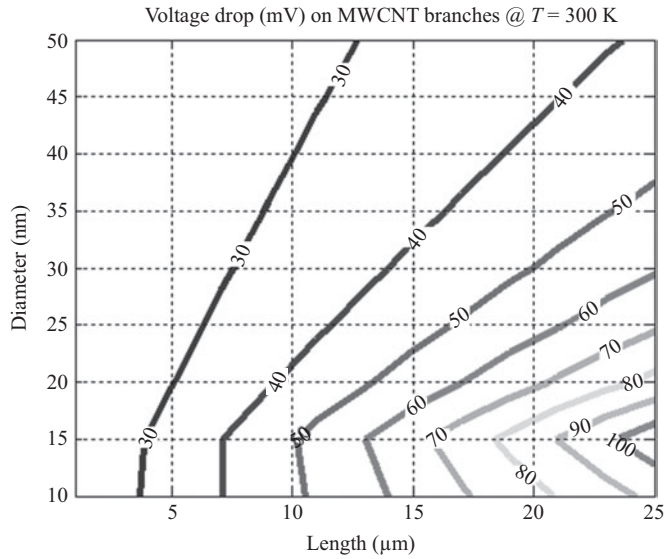


Figure 7.12 Voltage drop for MWCNT interconnects with different lengths and diameter for $T = 300\,K$

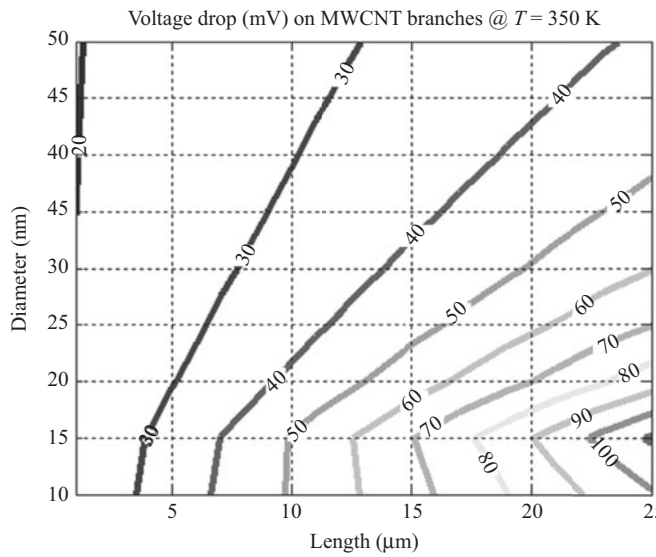


Figure 7.13 Voltage drop for MWCNT interconnects with different lengths and diameters for $T = 350\,K$

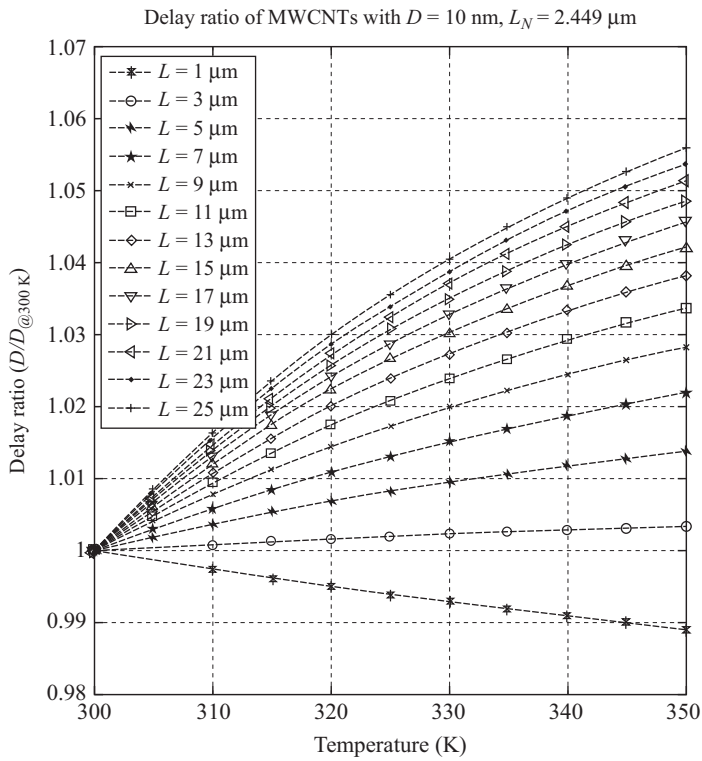


Figure 7.14 Delay for MWCNT interconnects with different lengths and diameter = 10 nm

nanotubes with diameter 10 nm, whereas for short-length nanotubes a delay speed up is obtained due to negative TCR effect. For nanotubes with large diameter, the changes in delay are minor which points that the nanotube dimensions that are immune to temperature effects.

7.4 Conclusion

CNTs due to their unique, thermal, and electrical properties are being investigated as promising candidate material for signaling interconnect and power/ground delivery networks. The attractive negative thermal resistance coefficient presents an advantage over other metal conductors for implementing reliable on-chip interconnects. In this chapter, we performed a detailed electro-thermal analysis of horizontally and vertically aligned CNTs by investigating the change in resistance due to temperature and its impact on interconnect performance and voltage drop. Analyses demonstrate that CNTs can be efficiently exploited for both signaling and power/ground delivery networks while carefully selecting their diameters and lengths for minimizing the thermal effects.

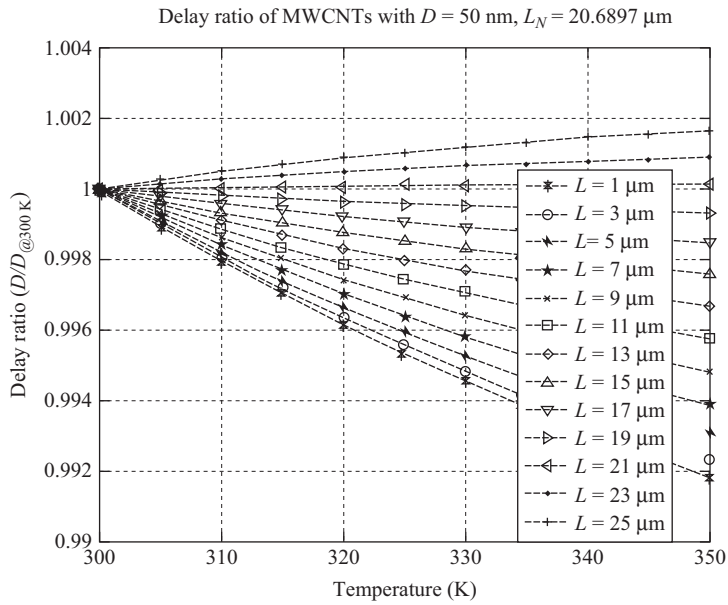


Figure 7.15 Delay for MWCNT interconnects with different lengths and diameter = 50 nm

References

[1] Li, H., Xu, C., Srivastava, N., Banerjee, K.: "Carbon nanomaterials for next-generation interconnects and passives: physics, status, and prospects," *IEEE Transactions on Electron Devices* **56**(9), 1799–1821 (2009). DOI: 10.1109/TED.2009.2026524

[2] Naeemi, A., Sarvari, R., Meindl, J.: "Performance comparison between carbon nanotube and copper interconnects for gigascale integration (gsi)," *IEEE Electron Device Letters* **26**(2), 84–86 (2005). DOI: 10.1109/LED.2004.841440

[3] Naeemi, A., Huang, G., Meindl, J.: "Performance modeling for carbon nanotube interconnects in on-chip power distribution," *IEEE Electronic Components and Technology Conference*, pp. 420–428 (2007). DOI: 10.1109/ECTC.2007.373831

[4] Li, H., Liu, W., Cassell, A., Kreupl, F., Banerjee, K.: "Low-resistivity long-length horizontal carbon nanotube bundles for interconnect applications 2014 – Part II – characterization," *IEEE Transactions on Electron Devices* **60**(9), 2870–2876 (2013). DOI: 10.1109/TED.2013.2275258

[5] Srivastava, N., Banerjee, K.: "Performance analysis of carbon nanotube interconnects for VLSI applications," *IEEE International Conference on Computer Aided Design*, pp. 383–390 (2005). DOI: 10.1109/ICCAD.2005.1560098

[6] Knickerbocker, J., Andry, P., Dang, *et al.*: "3d silicon integration," *IEEE Electronic Components and Technology Conference*, pp. 538–543 (2008). DOI: 10.1109/ECTC.2008.4550025

[7] Wilder, J.W.G., Venema, L.C., Rinzler, A.G., Smalley, R.E., Dekker, C.: "Electronic structure of atomically resolved carbon nanotubes," *Nature* **391**(6662), 59–62 (1998)

[8] Qin, L.-C., Zhao, X., Jirahara, K., *et al.*: "The smallest carbon nanotube," *Nature* **408**(6808), 50 (2000)

[9] Burke, P.: "Luttinger liquid theory as a model of the gigahertz electrical properties of carbon nanotubes," *IEEE Transactions on Nanotechnology* **1**(3), 129–144 (2002). DOI: 10.1109/TNANO.2002.806823

[10] Nieuwoudt, A., Massoud, Y.: "Evaluating the impact of resistance in carbon nanotube bundles for VLSI interconnect using diameter-dependent modeling techniques," *IEEE Transactions on Electron Devices* **53**(10), 2460–2466 (2006). DOI: 10.1109/TED.2006.882035

[11] Zhu, L., Sun, Y., Xu, J., *et al.*: "Aligned carbon nanotubes for electrical interconnect and thermal management," *IEEE Electronic Components and Technology Conference* vol. 1, pp. 44–50 (2005). DOI: 10.1109/ECTC.2005.1441243

[12] Khan, N., Hassoun, S.: "The feasibility of carbon nanotubes for power delivery in 3-D integrated circuits," *In: Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, Sydney, NSW, pp. 53–58 (2012). DOI: 10.1109/ASPDAC.2012.6165010

[13] Naeemi, A., Meindl, J.: "Physical modeling of temperature coefficient of resistance for single- and multi-wall carbon nanotube interconnects," *IEEE Electron Device Letters*, pp. 135–138 (2007)

[14] Aliev, A.E., Lima, M.H., Silverman, E.M., Baughman, R.H.: "Thermal conductivity of multi-walled carbon nanotube sheets: radiation losses and quenching of phonon modes," *Nanotechnology* **21**(3), 035709 (2010). DOI: 10.1088/0957-4484/21/3/035709

[15] Jiang, J., Saito, R., Grüneis, A., *et al.*: "Photoexcited electron relaxation processes in single-wall carbon nanotubes," *Physical Review B* **71**, 045417 (2005). DOI: 10.1103/PhysRevB.71.045417. URL http://link.aps.org/doi/10.1103/PhysRevB.71.045417

[16] Pop, E., Mann, D., Reifenberg, J., Goodson, K., Dai, H.: "Electro-thermal transport in metallic single-wall carbon nanotubes for interconnect applications," *In: Electron Devices Meeting, 2005. IEDM Technical Digest*. IEEE International, pp. 4 pp. 252–526 (2005). DOI: 10.1109/IEDM.2005.1609321

[17] Li, H., Liu, W., Cassell, A.M., Kreupl, F., Banerjee, K.: "Low-resistivity long-length horizontal carbon nanotube bundles for interconnect applications – part II characterization," *IEEE Transactions on Electron Devices* **60**, 2870–2876 (2013)

[18] Li, H., Srivastava, N., Mao, J.F., Yin, W.Y., Banerjee, K.: "Carbon nanotube vias: a reality check," *In: Electron Devices Meeting, 2007. IEDM 2007*. IEEE International, pp. 207–210 (2007). DOI: 10.1109/IEDM.2007.4418903

Chapter 8

High-level synthesis of digital integrated circuits in the nanoscale mobile electronics era

Anirban Sengupta[1] *and Saraju P. Mohanty*[2]

Digital integrated circuits (ICs) are the main workhorse of all modern consumer electronic systems. Digital ICs are much more complex and more closely follow the technology scaling as compared to the analog or mixed-signal ICs. For example, the transistor count can be in billions the device sizes at this point can be 14 nm FinFET in the digital ICs. However, the good news for digital ICs is that the digital designs have well-defined abstractions including system, architecture, logic. This Chapter is focused at the architecture level of the digital ICs. In particular, detailed discussions of high-level synthesis technique has been presented that can generate digital ICs. Trust of electronic systems that are used in day-to-day life is critical. This Chapter also discusses the HLS technique that can generate trusted digital ICs.

8.1 Introduction

The impact of consumer electronics such as mobile phones, digital cameras, digital television, and DVD/MP3 players is profound on our society. The central module of these products is a miniature size integrated circuit (IC) which finds wide spectrum applicability from kitchen appliances, to automobiles, to aircrafts, or to any embedded systems. The system in these aforesaid modern consumer electronics products is built as an Analog/Mixed-Signal System-on-chip (AMS-SoC) [45, 46] where the digital circuits are the main computational modules, while the analog or mixed-signal components are interfacing circuits, in a typical case. Therefore, proficient design of digital ICs has become the need of the hour as it serves as one of the significant driving factors of efficient system design in this current mobile electronics era (the various factors such as speed, power, reliability involved during design of handheld devices is as shown in Figure 8.1). The complexity of digital ICs in terms of number/size of transistors is quite large. However, the digital ICs have well-defined designs of abstractions such as system, algorithm, register transfer, and logic which through

[1]Computer Science & Engineering, Indian Institute of Technology, Indore, India
[2]Department of Computer Science & Engineering, University of North Texas, USA

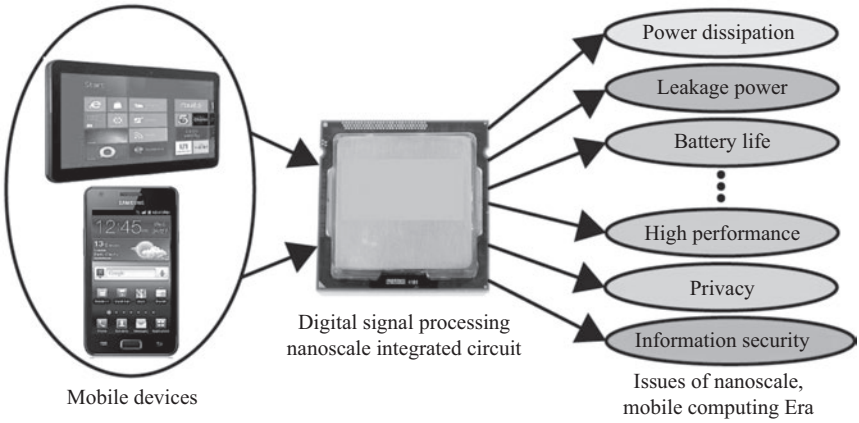


Figure 8.1 Factors involved in the design of handheld devices [57]

the application of divide and conquer approach can be exploited to handle the complex digital IC design flow. Each design abstraction layer has its own corresponding automated design methods or computer-aided design (CAD) methods which enables design of error free ICs within acceptable design time. One such automatic CAD technique is high-level synthesis (HLS; also known as behavioral, architectural, or algorithmic synthesis), which comprises design space exploration (DSE) process that allows exploration of design alternatives yielding to an optimized design option, prior to layout of the circuit in actual silicon. HLS is defined as the translation of a behavioral description to a structural description, i.e., from behavioral hardware description languages (HDLs) like VHDL, Verilog, SystemVerilog, to register-transfer level (RTL) VHDL and Verilog [14].

As discussed earlier, HLS is the transition of an application, represented through a control data flow graph (CDFG), from its system or algorithmic level description to the equivalent RTL counterpart while simultaneously satisfying conflicting user objectives/constraints such as area/power, delay (as shown in Figure 8.2 derived from Reference 47). HLS design process includes DSE which incurs convolution with the inclusion of ancillary variables related to loop manipulation during optimal scheduling of the CDFGs. These loop manipulation techniques (called high level transformations) are usually employed on the behavioral description early in the design cycle. The loop manipulation may comprise multiple ancillary variables such as 'loop unrolling', 'loop pipelining' and 'loop shifting'. The effects of applying these loop manipulation techniques are considered non-trivial and unforeseen [1, 6]. Though many of such aforesaid loop manipulation techniques are well known in the area of compiler optimization, however its' impact either jointly or discretely in the context of HLS optimization (tradeoff) remains subject of ongoing research and investigation. Studying the effect of loop manipulation techniques during power-delay tradeoff of datapath in HLS is especially crucial for the current generation of data/control intensive applications.

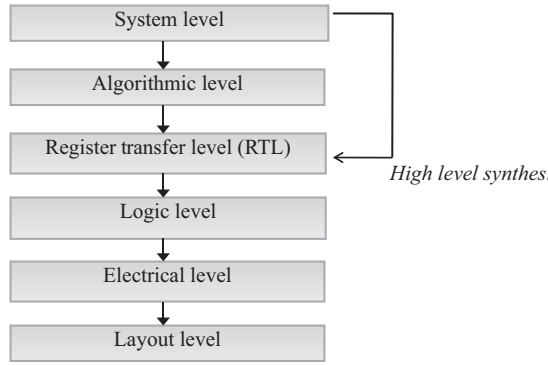


Figure 8.2 Abstraction levels of digital integrated circuit design

Besides above, current generation of hardware circuits is often vulnerable to hardware Trojans which are vindictive modifications to the design logic, made during either design stage or fabrication phase. This problem becomes in escapable owing to globalization involved in design process of system-on-chip (SoC) amassing major apprehensions of security and trustworthiness of embedded third party digital Intellectual Property (IP) cores (3PIPs) [32, 35]. An adversary can deliberately infuse a Trojan logic during the design stage of a 3PIP resulting in faulty digital circuit. This problem gets further complicated as hardware Trojans can be of various types [33, 34]. In the context of HLS, usually the hardware Trojan considered is the one which has capability to maliciously alter the digital output of a 3PIP core. Therefore, another critical aspect is accounting for the security aspect against untrusted third party digital IP cores during DSE in HLS. Considering hardware Trojan detection by an optimized secured datapath during HLS has not been done with absolutely zero effort so far in DSE of a user multi-objective constraint optimized hardware Trojan secured datapath [38]. This problem requires investigation as yielding a Trojan secured datapath is not frivolous.

For such aforesaid intricate and intractable problems, advanced DSE algorithms equipped with adaptive capabilities to change search direction, needs to be administered. Bacterial Foraging Optimization Algorithm (BFOA), Swarm Intelligence (SI), etc., are some of the advanced DSE frameworks that are known to perform well for intractable problems and are known to have the capabilities to search an optimal solution [1, 6]. Automated simultaneous exploration of loop manipulation factor and datapath for optimal scheduling based on power-delay tradeoff during HLS has been investigated in this chapter.

The rest of the chapter is organized as follows: Section 8.2 discusses the fundamentals of HLS with emphasis on some popular scheduling algorithms; Section 8.3 provides an overview on power aware HLS for nanoscale ICs with emphasis on the some recent methods as well as effects of loop manipulation on power and delay of the design; Section 8.4 discusses some selected bio/nature inspired heuristics for DSE framework with emphasis on a novel bacterial foraging driven DSE process;

Section 8.5 provides a detailed insight on secure information processing during HLS with emphasis on protection of design from external adversary for robust design output during HLS; Section 8.6 describes some well-known HLS tools; finally the conclusion of this chapter is provided in Section 8.7.

8.2 Fundamentals on high level synthesis

Any digital system can be classified into multiple levels of design abstraction. The layers of abstraction have been highlighted in Figure 8.2. In this abstraction level, the higher the layer, lesser are the lower level details available. However, greater are the chances of yielding dividends from optimization. On the contrary, the lower the layer, more details of low level information on circuits are available, however at the cost of yielding lesser chances of dividends on optimization. The ideal design strategy should be meet in the middle technology [56]. HLS plays a major role in enabling this meet in the middle technology for design of optimized digital circuits.

8.2.1 Overview on HLS design process

The process of HLS is analogous to a compiler which translates the high level language to its corresponding assembly language. A HLS design process converts the high level description of an application into its respective RTL circuit [16]. It consists of three main stages:

a. Scheduling: placing the operations in temporal domain, i.e., assigning operations of the CDFG to control steps.
b. Allocation: determining the number of instances of each resource needed for execution.
c. Binding: determining the hardware resource where the operation will be performed.

Interdependent tasks such as scheduling, allocation and module selection are important ingredients of the HLS design process. HLS is a methodology of transforming an algorithmic behavioral description into an actual RTL structure. Therefore HLS methodology contains a sequence of tasks to convert the abstract behavioral description of the algorithm into its respective structural block at register transfer (RT) level. The design at the RT level comprises functional units such as Arithmetic Logic Unit (ALU), storage elements, registers, buses, and interconnections. The algorithmic description specifies the inputs and outputs of the behavior of the algorithm in terms of operations to be performed and data flow. A description of the algorithm is usually represented in the form of an acyclic directed graph known as a sequencing graph. These graphs specify the input/output relation of the algorithm and the data dependency present in the data flow. The graph is defined in terms of its vertices and edges, where the vertices signify the operations and the edges indicate the data dependency present in the function. HLS is therefore a conversion from the

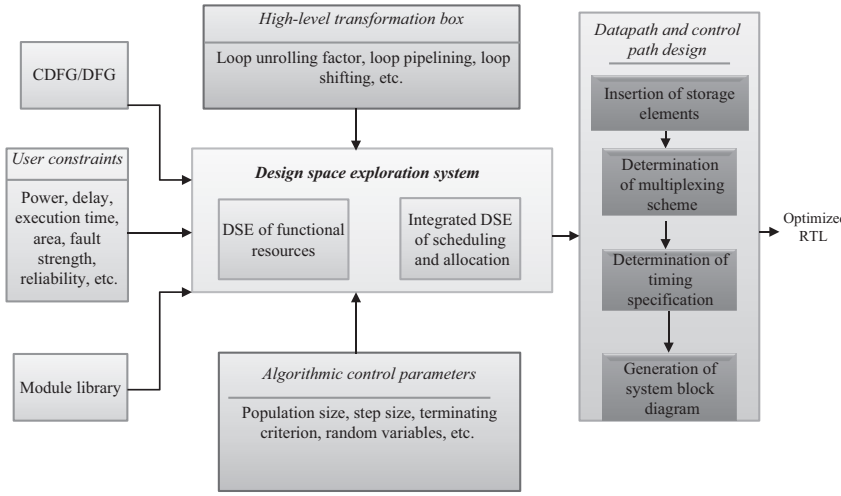

High-level transformation box
Loop unrolling factor, loop pipelining, loop shifting, etc.
CDFG/DFG
User constraints
Power, delay, execution time, area, fault strength, reliability, etc.
Design space exploration system
DSE of functional resources
Integrated DSE of scheduling and allocation
Module library
Algorithmic control parameters
Population size, step size, terminating criterion, random variables, etc.
Datapath and control path design
Insertion of storage elements
Determination of multiplexing scheme
Determination of timing specification
Generation of system block diagram
Optimized RTL

Figure 8.3 Various phases of high level synthesis for igital ICs

abstract behavioral description to its respective hardware description in the form of ALU, memory elements, storage units, multiplexers/demultiplexers, and the necessary interconnections. The transformed algorithm at the RT level is comprised of a control unit and the data path unit. HLS offers many advantages, such as productivity gains and efficient DSE. Performing DSE at a higher level of abstraction provides more dividend than at lower levels of abstraction, i.e., transistor level or logic level. Traditional HLS design methodology is much simpler than modern design techniques. In general, the initial step of synthesis is to compile the behavioral specification into an internal representation. The next step is to apply high level transformation techniques with the aim of optimizing the behavior as per the desired performance. In order to realize the structure, the final step is to perform scheduling to determine the time at which each operation is executed and the allocation, which is synthesizing the necessary hardware to perform the operations [47, 48].

The basic block diagram of a HLS design process is shown in Figure 8.3. As shown in Figure 8.3, it is evident that a HLS design process works with the following input modules: (a) CDFG/DFG (b) user constraints viz power, delay, reliability (c) module library (d) control parameters and I high level transformation box consisting of various transformation options such as loop unrolling, loop pipelining, loop shifting. The CDFG/DFG represents the behavior of the application, while user constraints are conflicting in nature that indicates the upper budget of the metric acceptable to the user. Further, module library comprises details of hardware resources such as speed, area, power. Control parameters generally comprise variables that define the sensitivity of the DSE algorithm such as in evolutionary schemes viz population size, step size, terminating criterion, random variables (however for non-evolutionary DSE algorithms, this module is mostly non-existent). This is mostly applicable for evolutionary (nature/bio-inspired) approaches. The high level transformation box generally

comprises information related to loop manipulation that modifies the CDFG. The DSE system yields either an optimized architecture (functional resource configuration, i.e., number of multipliers, number of adders etc) or an integrated structure of schedule along with its allocation information. This information is passed through the data-path and control path design block which comprises sub-steps such as insertion of storage elements for holding the data, determination of multiplexing scheme, followed by determination of timing specification per clock cycle basis and generation of system block diagram. Finally, the output of the HLS design process is an optimized RTL structure that implements the behavior (CDFG). The process of finding an optimized architecture solution that satisfies the user specifications is called 'design space exploration'. This DSE can be either an interdependent/integrated process of scheduling and allocation or discrete process of performing scheduling and allocation independently. Results have mostly been found to vary with the sequence/order of the steps adopted by the designer or automated tool during DSE [16].

The process of modern DSE is heavily reliant on user objectives of power and delay. Because of remarkable escalation in the demand of personal computing devices with limited battery life, power as a design objective has become the front runner of current research [17–19]. Generally, considering power during DSE in HLS is also motivated because of the following reasons:

- Sudden rise in emergence of power hungry portable devices.
- Temperature control/thermal consideration is possible with the optimization of power during HLS design.
- Reliability can be increased with the minimization of power dissipation. This is because, higher power consumption of devices leads to higher temperature which further leads to phenomenon such as electromigration and hot electron effect.

8.2.2 Need for HLS

Today's modern electronic systems are designed starting from specifications given at a very high level of abstraction. This is gradually becoming evident in many mainstream modern Electronic Design Automation (EDA) HLS tools which accept a design expressed in a high-level format as input and can automatically yield the respective RT/Logic/transistor-level implementation with very limited human involvement. All hardware systems can be classified into various levels of abstraction such as System level, Architecture level, RTL, Layout level, and Transistor level. Optimization performed at the higher levels of abstraction provides more flexibility, productivity and design specification awareness than performing only at the lower levels of abstraction. Moreover, although effective, performing optimization only at the transistor level is not sufficient for the current generation of high performance, power hungry application specific systems (used in embedded applications) due to the enormous complexity involved. The traditional method of optimization performed by circuit designers only at low level for area and latency is insufficient for current power and performance requirements. Therefore, the need for HLS has become extremely crucial. HLS consider user goals during the architecture selection process by performing optimization

of the given application based on high level parametric models. This would enable to generate a quality aware solution (at the RT-Level) with greater possibility of better optimization at the transistor level [47, 48].

HLS is an automated translation of the algorithmic behavioral level specification of the system to its respective RTL counterpart. Following are the list (but not limited to) of major benefits after deploying HLS to the designer and organization.

- Since more automation will exist with HLS, therefore, the product development will be faster and cheaper. This is because the design time will be faster compared to manual implementation at the RT-level.
- Ability to control the design architecture will be much higher due to availability of DSE that yields several design alternatives of equivalent functionality but with varying parametric tradeoff choices. This is a key feature in a good synthesis system where several design from the same specification can be yielded in a reasonable amount of time.
- More outreach to non-experts is possible with the automation that exists in HLS. This enables non-experts to become part of the chip design without being skilled in HLS.
- Design is specification aware from the very beginning of the design flow: This is because DSE to perform multi-objective optimization and tradeoff is needed from the very earliest stage of designing. This will enable the designers to start the development with an architecture that is already specification aware (high level optimized) from the highest level of abstraction thus rendering more chances that final design (logic/layout) corresponds to the given constraints [20].

8.2.3 Scheduling algorithms

Scheduling refers to the assignment of each operation to the control step, i.e., deciding the time sequence of each operation. The goal of the scheduling process is to determine the amount of time taken/number of control steps required/clock cycle required for executing a certain application. Scheduling is directly related to the amount of hardware resources available. In other words, if the amount of hardware functional resources is increased, scheduling will be faster, i.e., it will require lesser number of control steps to finish the task. On the contrary, if the hardware resources available are limited, time consumed by scheduling will be longer. Scheduling implementation can be either regular, chaining based, or multi-cycling based in nature. An example of scheduling for a sample application is shown in Figure 8.4. As shown in the Figure 8.4, the operations of the application (DFG) are divided into temporal domain such that the first time step executes two multiplications followed by addition and again two multiplications in second time step, keeping in mind any data dependency relation between operations. This process continues until all the operations of the DFG are exhausted. Finally, the delay of each control steps is summed up to estimate the total delay of the scheduling. It is important to understand, that the order of execution of operations (i.e., assigning time stamp) are driven through scheduling algorithms.

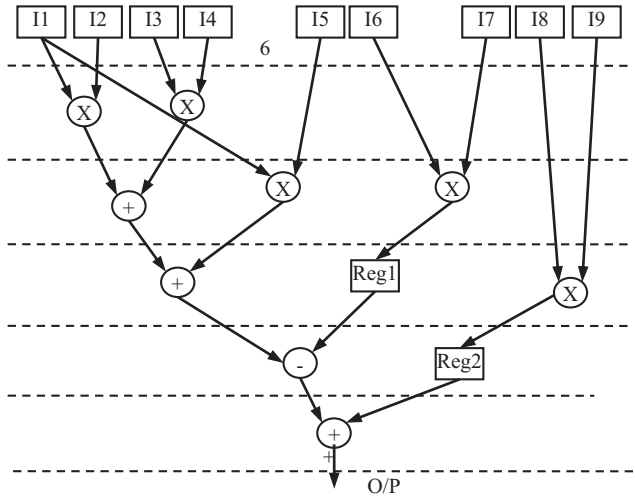


Figure 8.4 Example of scheduling

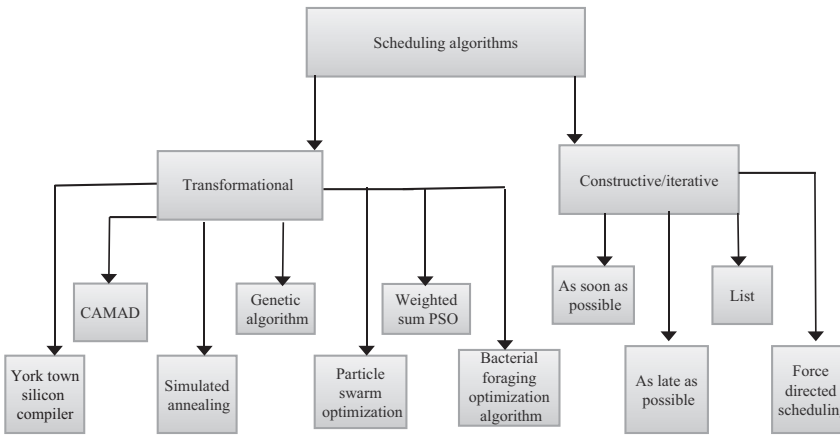


Figure 8.5 Selected scheduling algorithms used in HLS

Generally, scheduling algorithms can be classified into two categories [20, 21] as shown in Figure 8.5:

a. Transformational: This class of scheduling algorithms initiate with a default schedule and then apply transformations to generate new schedules. The initial schedule may be either a maximally serial implementation or a maximally parallel implementation. One of the well-known transformational scheduling algorithm is [20–22] which tried exhaustive combinations of serial and parallel transformations and selected the best

solution from that. This method, inevitably suffers from huge computational complexity and therefore does not fit at all for current large size applications. Besides above some of the other heuristic-based transformational scheduling algorithms that drives the solutions toward the user constraints/specifications: York Town Silicon Compiler (YSC) [27] and CAMAD design systems [28]. Further, modern heuristic driven scheduling algorithms are based on simulated annealing (SA) [29], genetic algorithm (GA) [3], weighted sum particle sum optimization (WSPSO) [51], discrete PSO [52], regular PSO [53], and BFOA [54]. In SA-based scheduling [29], the initial scheduled randomly generated acts as the seed solution. Then a set of efficient transformations are applied to generate new schedule solutions. These set of transformations are often called perturbations/moves. Each such solution generated through transformation are subjected to fitness evaluation based on power/area delay tradeoff. If the new solution is found to be better fit, then it is accepted straightaway, else the solution is accepted based on a certain probability that depends on a temperature control parameter which is lowered over time. This process repeats to yield the final schedule solution until the terminating criterion is reached. Moreover, GA-based scheduling [3], employs robust search techniques where each seed chromosome represents the list of operations with their respective workload/load factor. The workload/load factor encoded in each chromosome acts as the priority function to decide the preference during scheduling in case of resource conflict. On each chromosome, perturbations in the form of crossover and mutation is applied to generate a set of new chromosome which on decoding results into a valid scheduling solution. Each such decoded solution is subjected to fitness evaluation based on power/area-delay tradeoff. The best set of individuals are taken forward in the next generation for further evolution through genetic operators. This process repeats to generate the final schedule solution until the terminating criterion is reached. Further, in WSPSO [51], authors have integrated GA and PSO to evolve with a scheduling solution where a fitness is evaluated through power-delay based cost function. However, the authors have not used the mathematical velocity function to mimic exploration drift during DSE in their approach. Moreover in [52], PSO was used to perform DSE, however with no consideration on social factor during evolution. Authors [53], performed full mapping of PSO variables to design DSE of hardware units using velocity function as exploration drift. The optimized architecture was used as allocation hardware available for driving the scheduling. BFOA in [54] used tumble vector to change direction during searching for an optimal solution and was driven by chemotaxis and dispersal function to evolve with an optimized architecture for generating the scheduling.

b. Constructive/Iterative: This class of scheduling algorithms generates the schedule by adding one operation at a time. This process continues until all the operations have been scheduled. Mostly the next operation to be chosen is made based on some priority function driven through the need of area minimization, peak power minimization, etc. Some of the traditional algorithms in this category are As Soon As Possible (ASAP), As Late As Possible (ALAP), List scheduling, and Force-Directed Scheduling (FDS). ASAP scheduling only requires the information on the number of instances of each resource available. It yields the schedule with the least

latency possible with a given resource constraint. The operations are topologically sorted based on dependency information between them and then the operations are placed in the control step based on the hardware resource available. On the contrary, ALAP schedule is performed under fixed latency constraint. The latency constraint may be derived from the latency of the ASAP schedule, otherwise if the latency is over constraint, then, no solution may exist. In this algorithm, the operations are deferred as late as possible without exceeding the latency constraint. In both the above scheduling techniques, no consideration is given to the resource constraints and no priority is given to nodes on critical path. This may result in a less critical node being scheduled ahead of critical node. The aforesaid limitation has been taken care in list scheduling which is based on a global node selection criteria. This algorithm maintains an ordered list of operations to be scheduled in each control step based on the following two conditions: (a) its predecessors are scheduled and (b) the resource to compute is available. The order of operations in the list is achieved through some priority function. The priority function can be the length of the path from the operation to its end block. However various priority function can be used to decide the urgency of the operation to be scheduled earlier. Another type of scheduling algorithm is mobility-based scheduling where both the ASAP and ALAP schedules are computed. Next for each operation, its respective range between ASAP and ALAP schedule is determined. The difference in control steps for each operation obtained from ASAP and ALAP is called mobility. Here the priority function is inversely proportional to the mobility, i.e., the operation with higher mobility is scheduled later than the operation with lesser mobility. This generates an optimal schedule between ASAP and ALAP. Mobility-based scheduling can also be considered a type of list scheduling. Besides above, FDS is a well-known scheduling approach which is applied to time constrained problems. The algorithm targets to minimize the number of resources (functional units) required during scheduling. In FDS, a distribution graph is created based on the ASAP and ALAP graph. For each operation, force value is calculated using certain functions based on probability. When the total force of all operations are calculated, the operation with the least force is scheduled in that control step. The algorithm does not guarantee to yield optimal solution always. The FDS algorithm terminates when all the operations of the CDFG are scheduled [20–22].

8.2.4 Allocation and binding

It is a process of assigning operations/variables of the CDFG to hardware units and registers based on availability. The availability of hardware and storage elements is usually checked from the module library available for the design. It is also possible that different hardware for similar operations are available in the module library in which case, local heuristics may be employed to decide the hardware unit for the operation. The goal of the allocation process is to minimize the usage of hardware functional units and registers required by the design. The process of minimization of hardware units is usually performed by resource sharing, i.e., similar operations of a CDFG executing/scheduled in different control steps (mutually exclusive operations) are assigned to same hardware unit. Similarly, register (memory) minimization is performed by

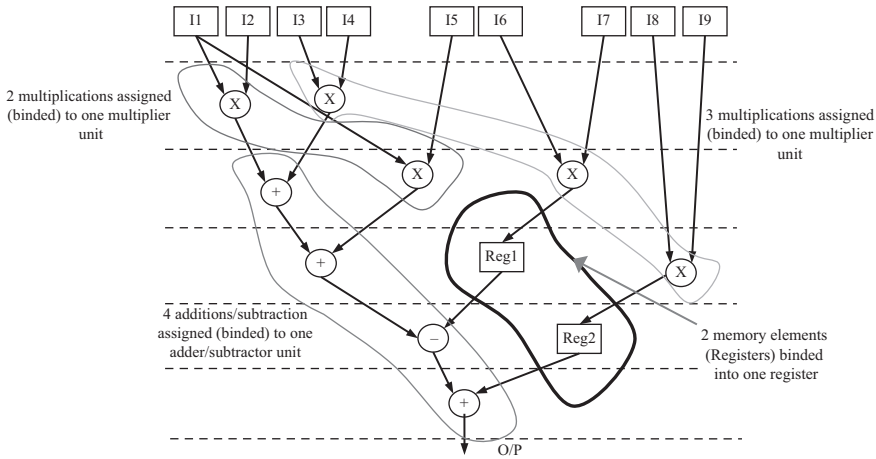


Figure 8.6 Example of allocation/binding

register sharing, i.e., two different registers are physically binded to the same/single register unit. This process incurs an overhead of extra wiring and switching devices due to sharing of resources. Nevertheless, it results in reduced chip area. Therefore, the process of allocation/binding is the mapping of CDFG operations to functional resources, assigning values to registers and providing interconnections between hardware units and registers [20, 21, 47, 48]. An example of allocation/binding is shown in Figure 8.6.

8.3 Power, energy, or leakage aware HLS for nanoscale ICs

Consideration of design metrics such as power and energy during design of nanoscale ICs has been a subject of great interest in the VLSI/CAD community since last few decades. However, its contemplation during HLS with respect to design of nanoscale ICs has generated interest in the VLSI/CAD community since the last decade only, with major works arising during this period. This is because the metrics of power and energy have become extremely critical elements for optimization in this current mobile electronics era. This section will highlight some of the selected power/energy aware HLS approaches. The different power/energy aware HLS approaches in terms of target domain have been briefed in Figure 8.7. Additionally, the power/energy aware HLS approaches in terms of technique employed for power optimization is shown in Figure 8.8.

8.3.1 Selected power, energy, or leakage aware HLS methods

The problem of minimizing the power dissipation was addressed by considering allocation and binding by iterative improvement of some initial solution [23]. By using a Genesis behavioral synthesis system, concurrent register and module allocations were performed in order to reduce to interconnect. The approach uses the concept of

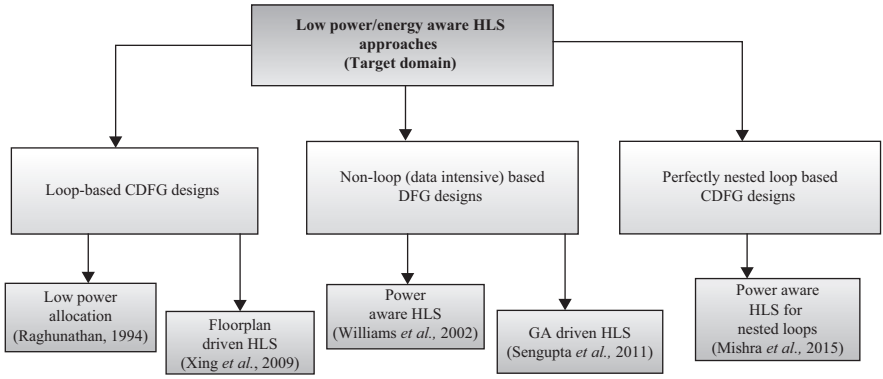

Low power/energy aware HLS approaches (Target domain)
Loop-based CDFG designs
Non-loop (data intensive) based DFG designs
Perfectly nested loop based CDFG designs
Low power allocation (Raghunathan, 1994)
Floorplan driven HLS (Xing et al., 2009)
Power aware HLS (Williams et al., 2002)
GA driven HLS (Sengupta et al., 2011)
Power aware HLS for nested loops (Mishra et al., 2015)

Figure 8.7 Various types of low power/energy HLS approaches classified in terms of target domain

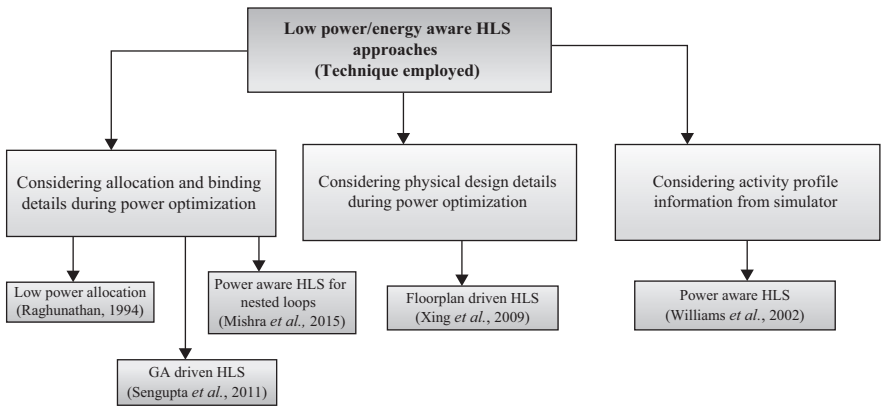

Low power/energy aware HLS approaches (Technique employed)
Considering allocation and binding details during power optimization
Considering physical design details during power optimization
Considering activity profile information from simulator
Low power allocation (Raghunathan, 1994)
Power aware HLS for nested loops (Mishra et al., 2015)
GA driven HLS (Sengupta et al., 2011)
Floorplan driven HLS (Xing et al., 2009)
Power aware HLS (Williams et al., 2002)

Figure 8.8 Various types of low power/energy HLS approaches classified in terms of techniques employed to optimize power

compatibility graph to analyze the lifetime of the operations. Compatibility between two operations exists when two operations can be alive at the same time and they do not need to be performed at the same time. Weights are assigned to the edges in the graph which is broken in case of a tie based on the switching activity weights. The allocation scheme selects the sequence of operations for a module or register such that the switching activity is reduced. Further, capacitance is reduced with the minimization of modules, register, and multiplexers.

The effect of binding during power optimization has been considered in Reference 5. The authors have used heuristic-based decoding technique to obtain valid schedule during GA driven DSE. The chromosomes are encoded with the hardware resource information and workload values of each operation. Using the information of the chromosome, a valid scheduling is derived using the decoding heuristic.

From each such valid schedule, the binding information yielding to the type/quantity of multiplexer/demultiplexer is extracted. Power of such multiplexer/demultiplexer units are evaluated along with the power of the hardware resources using the static power function and fed to the fitness block. This process evolves to yield a power aware schedule during HLS.

A particle swarm optimization (PSO) process has been deployed that considers the effect of power on the resultant design for perfectly nested loop based CDFGs during HLS through a power estimation model [49]. The process encodes the values of unrolling factor array into the particle position along with resource configuration. The fitness function of the DSE process during evaluation considers dynamic and static power during loop unrolling. The average dynamic power is a function of dynamic energy and execution delay which is estimated for the unrolled scheduling based on the explored resource configuration and nested loop unrolling factor. The effect on multiplexer/demultiplexer due to nested loops is considered in their approach during evaluation of resultant power. The process of exploration repeats until, the final power aware solution indicating the optimal unrolling factor array and datapath configuration that satisfy the power and delay constraint is yielded. Figure 8.9 shows an example of nested loop unrolling scheduling for autocorrelation benchmark in [49] which is used in estimating power of the design. In Figure 8.9 (derived from Reference 49), C^{G}_{first} and C^{G}_{cycle} are delay of first group and cycle time between consecutive groups, respectively (delay is multiplier is assumed to be 550 control steps and delay of adder is 14 control steps, respectively). Improvements in QoR was obtained for [49] when compared to recent literature.

An integrated power aware behavioral synthesis system that does not reply on techniques such as turning off idle parts of the system, or a controlled reduction in power supply was employed [50]. Power is estimated by utilizing activity profile from the simulation of the design on any standard HDL simulator. This data along with the information of area and delay guides the optimization process of power aware design. Features such as supply voltage scaling, pipelined and multi-cycle units are also considered in order to yield power aware design. The approach is further capable to generate multiple structural implementations of the same design with varying power/delay/area tradeoff. Healthy improvements in the reduction of energy consumption were obtained.

Power aware HLS is considered by considering interconnects information and physical lower level details through Floorplan driven multi-voltage synthesis [55]. This allowed reduction in the gap between HLS and physical level synthesis. More specifically, a major physical design step called 'floorplanning' is integrated into HLS process during evaluation. Since interconnects incur approximately around 20% of the total power in the circuits; therefore it was also considered during HLS design evaluation. Owing to the quadratic impact of voltage on power of the design, this work utilizes the advantage of lowering the voltage of hardware resources which do not lie on the critical path, nevertheless at the expense of incurring higher delay. The approach takes as its input a DFG, a component library, and constraints, such as resource constraint, delay constraint, clock cycle time, and the voltage domain (available supply voltages). After an initial scheduling and binding is performed, the

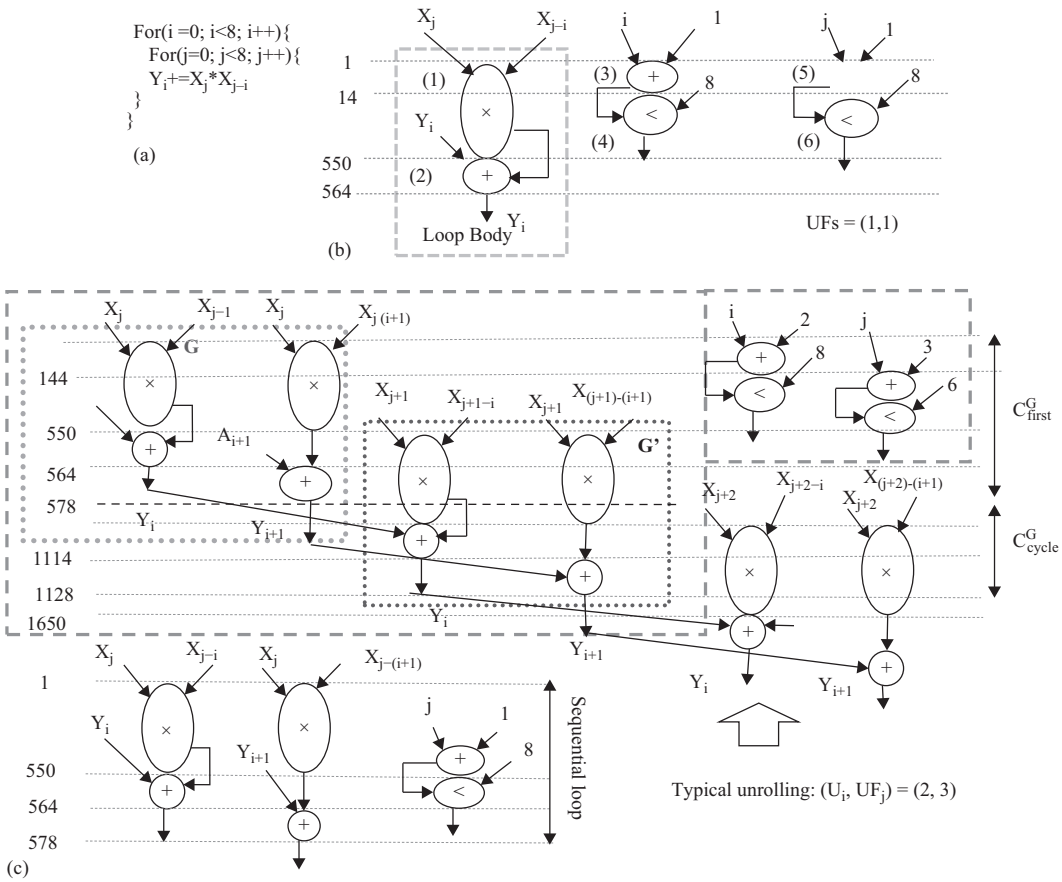


Figure 8.9 (a) 'C' code for original loop of Autocorrelation benchmark. (b) CDFG of original loop. (c) Determine and for G where G'. Where G and G' formed by scheduling twice 'λ' time unrolled loop body with ASAP with 2(*), 1(+), 1(<) (The portion in green box is only unrolled during execution time determination through our model), complete figure shows typical physically unrolled loop for UF = (2,3) [49]

algorithm starts an SA process. In each iteration, of the process, a floorplanner, a power/energy estimator, and local perturbation moves are performed in sequence, until a satisfactory synthesis solution is obtained. The SA process produces an optimized result for scheduling, binding as well as floorplanning. The cost function used is the total power/energy consumption by modules, registers, MUXes, level converters, and wires. The perturbation moves applied are as follows: (a) changing the voltage assignment of a resource, (b) changing the resource binding, (c) changing the operation schedule, (d) hardware resource swap. The generic flow of floorplan driven multi-voltage synthesis is shown in Figure 8.10 [55].

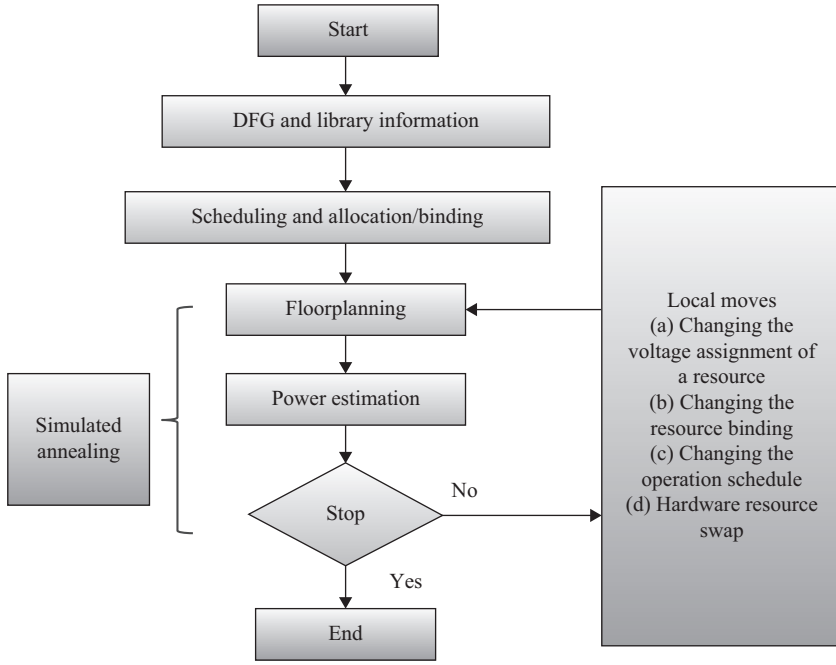


Figure 8.10 The flow diagram of floorplan driven multi-voltage synthesis [55]

8.3.2 Effects of loop manipulation on power and delay of the design

As discussed earlier, effects of loop manipulation in the context of HLS is non-trivial. This is because loop manipulation when employed during DSE involves consideration of its impact on the power and delay of the schedule. Loop unrolling involves duplication of the loop body multiple times to exploit additional parallelism across loop iterations [24]. Loop unrolling leads to performance improvement due to execution of multiple iterations of loop body in parallel. However, the adverse effects of loop unrolling are the increase in control states and code density. In other words, the size of the DFG increases indirectly impacting the performance of the design. Furthermore, due to unrolling of loop, code size increases which has negative impact on performance during scheduling. Additionally, when the datapath resource configuration is varied, the same loop unrolling may produce multiple alternatives of scheduling. On the other hand, for a fixed datapath resource configuration, multiple loop unrolling factors lead to various schedule alternatives. Loop unrolling also leads huge operation sharing leading to greater multiplexer/demultiplexer size. This increase of multiplexer/demultiplexer size leads to more power consumption and larger delay. If the delays through the controller and multiplexers/demultiplexers are not accounted for then the schedule resulting during HLS optimization can easily be imprecise. An example of loop unrolling for FIR benchmark for a specific resource

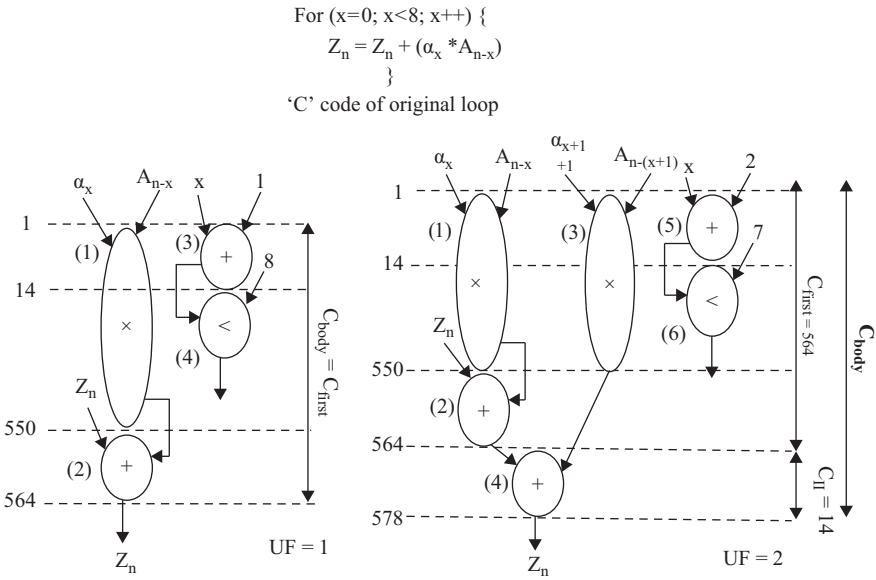


Figure 8.11 Demonstration of loop unrolling based on a resource constraint of 2(*), 2(+), 1(<) for FIR [15]

configuration is shown in Figure 8.11 (derived from Reference 15). where, 'C_{body}' is the number of CS required to execute loop body once, 'C_{first}' is number of CS required to execute first iteration, 'I' is the maximum loop iteration (I = 8), and 'C_{II}' is the number of CSs required between initiations of consecutive iterations.

a. Execution delay: The execution delay prediction model for a loop unrolled CDFG is derived considering the following three possible cases [1, 15]:

Case 1: When the unrolling factor (UF) is equal to one (indicates no unrolling) then,

$$\text{For } (x = 0; x < 8; x{+}{+}) \{Z_n = Z_n + (\alpha_x * A_{n-x})\}$$

'C' code of original loop

*Total # of control steps (CSs) = # of CSs required to execute loop body once * # of duplicate iterations of loop body* [1]:

$$C_T = (C_{first} * I) \tag{8.1}$$

C_T and α are total CSs required to execute the loop completely and $\left(\frac{I}{UF}\right)^{\text{quotient}}$.

Case 2: When UF evenly divides the loop count (I), then the total number of CSs is [1]

$$C_T = (C_{first} + (UF - 1) * C_{II}) * \alpha \tag{8.2}$$

Case 3: When UF unevenly divides I: in such a case, I mod UF iterations will be executed sequentially, therefore, the total number of CSs is [1]

$$C_T = (C_{first} + (UF - 1) * C_{II}) * \alpha + (I \bmod UF) * C_{first} \tag{8.3}$$

{Total CSs for unrolled loop} {Total CSs for sequential loop}
Furthermore, execution time for the system calculated as [1]

$$T_E = \Delta * C_T \tag{8.4}$$

Loop unrolling has further complications when it comes to the choice of its optimal UF during DSE. This is because performance does not monotonically increase with increase in UF. This is because the unroll factor may not evenly divide the iteration count resulting in a trailer loop in which the remaining iterations are sequentially executed, which affects the overall latency adversely [24]. Though the control states are an increasing function of the increase in unroll factor value, however, the delay as mentioned is a more complicated variable. Further, large unroll factors, though decreases the clock cycle of the design however, are not known to provide the best returns. From literature [24], it has been established that the performance improvement is found to be marginal for relatively large unroll factors. This may be because there may not be adequate resources available to exploit the huge parallelism due to large value of unrolling. Besides, large UF leads to huge increase in sharing of operations, affecting the power of the design at minimal improvement in performance. The resulting design may not be a fit candidate for exploration. Concepts of choosing a threshold value for UF during exploration thereby joins the decision making process. Techniques such as the one in Reference 2 that directly discard UFs which are nondivisible to the iteration count are considered inefficient because of the chances of losing an eligible unrolling candidate which may lead to an optimal solution. Subsequently, screening mechanisms needs to be devised to eliminate some candidates with large UF during DSE.

Additionally, loop manipulation also comprises techniques such as loop pipelining that improves performance. Loop pipelining is a process of initiating the next iteration of the loop before the previous iteration completes. This can be achieved through 'resource constrained software pipelining'. In this process, the loop body is unrolled completely (i.e., duplicating the loop body the maximum permissible time according to the iteration count), followed by resource constrained scheduling until a pattern emerges. Once, the pattern emerges, the scheduling is no longer required to be performed. Then initiation interval (or cycle times) is determined and total delay is evaluated. An example of resource constrained loop pipelining for FIR is shown in Figure 8.12. In the figure of loop pipelining provided, scheduling is only performed until a pattern emerges. Here the pattern is noticed between a set of every two consecutive iterations, i.e., i and $i+1$ (group G1) and $i+2$ and $i+3$ (group G2). The delay of the first group is denoted by L^G and the cycle time (difference between two consecutive groups G1 and G2) is denoted by T_C^G. This pattern enables to predict the total delay or execution time of the CDFG without requiring to tediously schedule

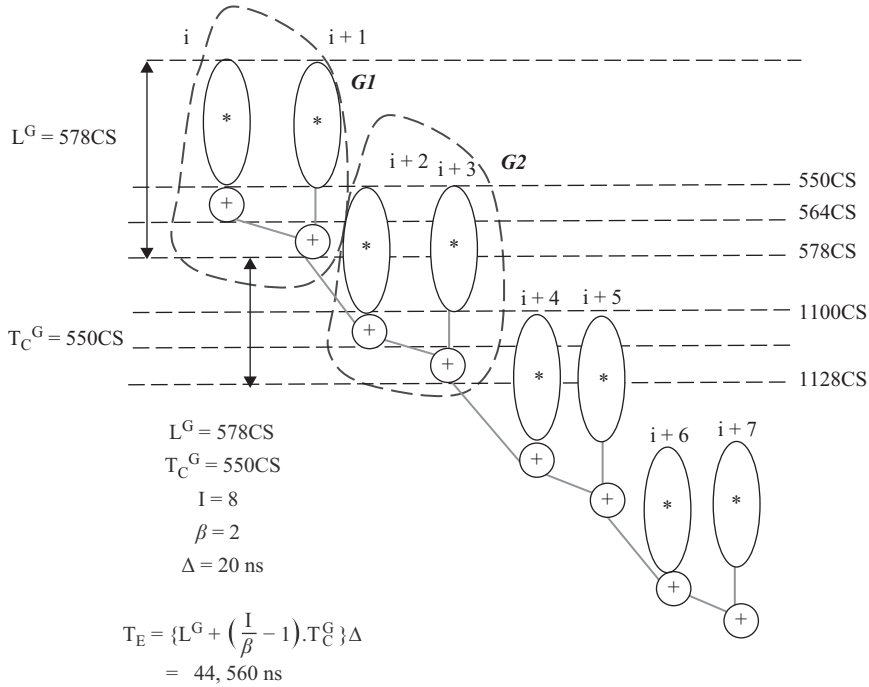


Figure 8.12 Resource constrained loop pipelining based on 2 (*), 2 (+), 1 (<) for FIR with maximum eight iterations

remaining iterations viz. i + 4, i + 5, i + 6, i + 7. Similarly, another example of loop pipelining for FIR based on single instance of multiplier is shown in Figure 8.13. Clearly due to lack of enough hardware resources, higher degree of parallelism could not be exploited in this case. For the sake of demonstration, multiplier hardware and adder/subtractor are assumed to consumed 11,000 ns and 270 ns, respectively (where 1CS = Δ = 20 ns).

The prediction of the total delay or execution time of the CDFG is performed using the following loop pipelining model:

$$T_E = \left\{L^G + \left(\frac{I}{\beta} - 1\right) \cdot T_C^G\right\} \Delta \tag{8.5}$$

where L^G, T_C^G have been defined earlier while I = total number of iterations of the loop, β = number of hardware instances corresponding to the maximum parallel independent operations, and Δ = duration of one CS during operation chaining based scheduling. However, the above prediction model does not hold valid when: I Mod $\beta \neq 0$. In such cases, the execution delay is estimated by completely scheduling

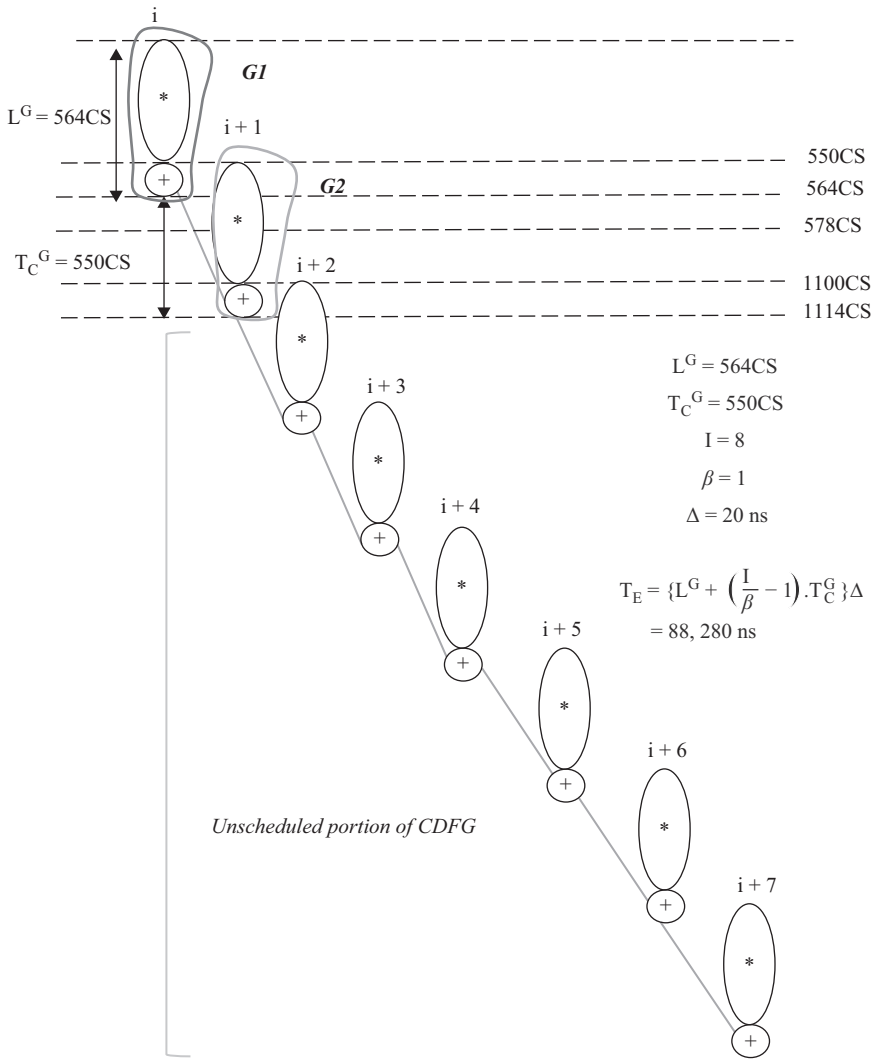


Figure 8.13 Resource constrained loop pipelining based on 1 (*), 2 (+), 1 (<) for FIR with maximum eight iterations

the fully unrolled loop body. *Note: the pattern emergence occurs after scheduling operations for '2β' loops. For example in Figure 8.10, β = 2, therefore, pattern is detected after scheduling loops till 2β, i.e., 4 iterations.*

Loop pipelining (also called software pipelining [25]) may be considered as a special case of loop unrolling where the UF is equal to the maximum iteration count. However like loop unrolling, loop pipelining also impacts power of the schedule. Therefore, considering loop pipelining and loop unrolling both during DSE of

CDFGs becomes critical during power-delay tradeoff. This motivates the need to devise intelligent methodologies that are capable to explore an optimal combination of loop unrolling/loop pipelining and datapath resource configuration during HLS that satisfies the user power-delay budget.

Another type of loop manipulation technique widely known in the area of HLS is called 'loop shifting' (or loop compaction). Loop shifting exploits the advantage of code compaction as well as parallelization of loop body, thereby enables to obtain shorter schedule length. However, loop shifting comes with a disadvantage of operation overhead. Nevertheless, the overall performance gain obtained due to large enough execution of loop body, compared to the overhead of loop operation is always greater for class of applications belonging to the multimedia and image processing. The process of loop shifting works as follows: an operation is shifted from the beginning of the loop body to the end. Therefore, to preserve rightness of the code, the copy of the shifted operation is inserted in the loop head of the code (stored with a different variable name). Finally, code compaction is performed which reduces the schedule length. The effect of loop shifting due to insertion of a copy of the shifted operation increases the code density thus impacting power of the design. However, this process enables substantial savings of execution delay if the loop body is executed adequate number of times [7, 26].

b. Power: The power estimate for loop unrolled design may be made using the following models:

The total power (P^T) from [1, 6] is

$$P^T = P^S + P^D \tag{8.6}$$

where, static power component is represented by P^S and average dynamic power is represented by P^D of a candidate solution.

Static power is calculated by the formula [1, 6]:

$$P_S = \left[\sum_{i=1}^{v} (N_{R_i} \times K_{Ri} + N_{MUX\,/\,DMUX} \times K_{MUX\,/\,DMUX} \right] * P_c \tag{8.7}$$

where 'N_{Ri}' indicates the number of instance of resource R_i, 'K_{Ri}' indicates the area utilized by resource R_i, 'v' is the number of resource types, '$N_{MUX/DMUX}$' is count of the multiplexer/demultiplexer, '$K_{MUX/DMUX}$' is area utilized by a multiplexer/demultiplexer, and 'p_c' indicates the power dissipated/per unit area (e.g., transistors) [1, 6]. Average dynamic power is calculated by from [1, 6]

$$P_D = \frac{\alpha * (E_{FU} + E_{MUX/DMUX})}{\Delta * ((C_{first} + (UF - 1) * C_{II}) * \alpha + (I \bmod UF) * C_{first})} \tag{8.8}$$

E_{FU} and $E_{MUX\,/\,DMUX}$ are the energy expended by resources and multiplexer/demultiplexer, respectively.

While, the average dynamic power estimate for loop pipelined design may be made using the following:

$$P_D = \frac{\alpha * (E_{FU} + E_{MUX/DMUX})}{\{L^G + \left(\frac{1}{\beta} - 1\right) \cdot T_C^G\}\Delta} \tag{8.9}$$

8.3.3 Other design space exploration approaches during HLS

Automated integrated exploration of datapath resource configuration and loop manipulation factor (unrolling or pipelining) for optimal operation chaining based scheduling during HLS remains a subject of investigation in VLSI/CAD community. There have been many works on DSE in HLS, some of which are mentioned herewith (refer Figure 8.14 for various DSE approaches in terms of the optimization goals). DSE of CDFGs was addressed based on multi-objective user constraints with focus on searching an optimal datapath resource configuration and UF [1]. GA framework was employed for solving the DSE problem based on area-delay tradeoff where manual intervention is required to decide UF. The work also considers only evenly divisible UFs as potential candidates ignoring other potential candidate UF during DSE [2]. Further, GA framework is again used for exploring the design space comprising datapath resource configuration as candidates, however it is not capable of executing exploration of optimal loop UF of CDFGs for optimal scheduling [3]. A scheduled data flow graph is accepted as an input in Reference 4 during DSE which does not aim to resolve the scheduling problem as well as optimize loop manipulation factor for CDFGs. The approach is based on a cost function which only considers functional resources ignoring resource binding (area of MUX and DMUX) information [4]. Exploration of optimal scheduling using GA framework was performed for data flow graphs with no focus on loop-based CDFGs [5]. BFOA is used for exploration of datapath resource configuration based on power-delay tradeoff, however with no focus on optimization of loop unrolling with various candidate UFs or loop pipelining [6]. A tool for HLS which selects potential UF candidates for the loop through user-directed control is employed [7]. Further, a modified fast SA was employed based on decision tree machine learning algorithm for performing multi-objective DSE in HLS. The approach is faster than standard SA however; with no focus on simultaneous exploration of datapath and unrolling for optimal chaining based scheduling [10]. A technique based on learning-based method was employed which used prediction-based method to evaluate solution quality. This enables to improve the solution quality as well as speed compared to local search techniques such as GA and SA. However it does not aim at integrated exploration of UF and datapath resources for loop-based CDFGs [11]. Additionally, in the context of DSE with HLS, learning-based methods were employed [12, 13]. These works rely on either local-search techniques [12] or common learning models [12, 13]. However, the above works do not focus on automated exploration of optimal loop unrolling and datapath resource configuration.

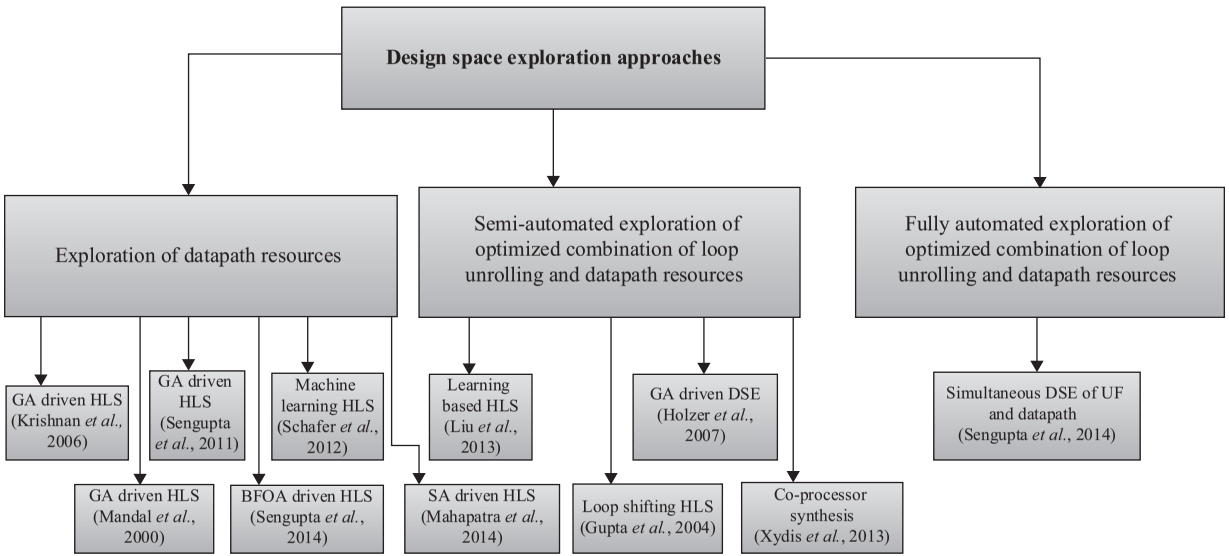

Design space exploration approaches
Exploration of datapath resources
Semi-automated exploration of optimized combination of loop unrolling and datapath resources
Fully automated exploration of optimized combination of loop unrolling and datapath resources
GA driven HLS (Krishnan et al., 2006)
GA driven HLS (Sengupta et al., 2011)
Machine learning HLS (Schafer et al., 2012)
Learning based HLS (Liu et al., 2013)
GA driven DSE (Holzer et al., 2007)
Simultaneous DSE of UF and datapath (Sengupta et al., 2014)
GA driven HLS (Mandal et al., 2000)
BFOA driven HLS (Sengupta et al., 2014)
SA driven HLS (Mahapatra et al., 2014)
Loop shifting HLS (Gupta et al., 2004)
Co-processor synthesis (Xydis et al., 2013)

Figure 8.14 Different types of DSE approaches

8.4 Bio/nature-inspired algorithms for DSE framework

DSE is widely known to a highly notorious optimization problem owing to multiple dimensions of complications involved in it. Therefore, to address such an intractable problem, bio/nature-inspired heuristic algorithms have proven to yield excellent results owing to the stochastic nature of its framework. Bio/nature-inspired algorithms generally mimic the nature or biological inspired process to escape local minima, thereby yielding high quality optimization results during DSE. This section will discuss the most recent and popular bio/nature inspired algorithms employed for DSE framework.

8.4.1 Selected bio/nature-inspired approaches

a. Bacterial foraging optimization algorithm (BFOA): BFOA is a new bio-inspired algorithm which has gained a lot of recognition in multiple problems of optimization. The reason for choosing BFOA optimization problems is its flexibility and adaptive nature. It is not too rigid and has very few factors that dictate the search criteria unlike in the case of GA where the search is not as adaptive as in the case of BFOA. Unlike GA and PSO, in BFOA, if a path is found unproductive, it has easy adaptation techniques such as change in direction to travel in a different path and elimination dispersal (ED) in the prospect of reaching the global best and not getting stuck at local optima. BFOA also has technique to eradicate premature convergence by increasing the step-length. The BFOA consists of tumble which dictates the direction in which it has to travel and swim to move in the chosen direction. These are the two mechanisms that dictate the locomotion of the bacterium. PSO, on the other hand, is considered a very sensitive algorithm; where inability to clinically pre-tune the PSO variables often may result in convergence to local optima [9]. BFOA, thus, is intuitively better to solve optimization problem which is considered to be intractable and NP hard. The algorithm is designed for continuous function optimization problem domains. The major processes in BFOA which are used to derive new solutions are called chemotaxis and dispersal. The chemotaxis function used for generating new solutions (bacterium position) in the form of hardware resource array (i.e., number of instances of adders, multipliers, subtractors etc) during DSE is [30]

$$x_i^{New} = x_i^{Last} + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{8.10}$$

where x_i^{New} = new resource set of i^{th} bacterium solution, x_i^{Last} = last resource set of i^{th} bacterium solution, $C(i)$ = step size and Δ is a random vector whose elements lie in $[-1, 1]$. Figure 8.15 shows the bacterium movement process in BFOA-DSE.

b. Particle swarm optimization: PSO is a heuristic search methodology that tries to imitate the travels of a flock of birds aiming at finding food [31]. PSO is based on a population of particles flying through a multi-dimensional search space. Each particle possesses a position and a velocity; both variables are changed to emulate

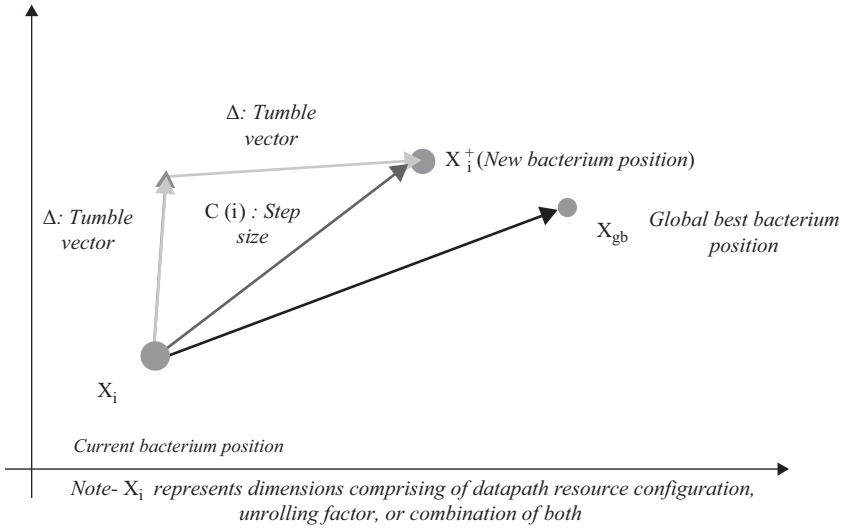

Δ: Tumble vector
X i+ (New bacterium position)
Δ: Tumble vector
C (i) : Step size
Xgb
Global best bacterium position
Xi
Current bacterium position
Note- Xi represents dimensions comprising of datapath resource configuration, unrolling factor, or combination of both

Figure 8.15 Bacterium movement in BFOA-DSE process

the social psychological tendency to impersonate the success of other individuals in the population. More formally, the position of i^{th} particle is changed by adding the velocity to the current position as follows [1]:

$$V_{id}^{+} = \omega V_{id} + b_1 r_1 [R_{lbi} - R_{id}] + b_2 r_2 \left[R_{gb} - R_{id}\right] \tag{8.11}$$

While, the new resource configuration is R_{id}^{+}, calculated using (8.12) [1]:

$$R_{id}^{+} = f(V_{id}^{+}, R_{id}) \tag{8.12}$$

where ω is called the inertia weight, b_1 is the cognitive learning factor, b_2 is the social learning factor, r_1 and r_2 are random numbers in the range [0, 1], R_{id} is the resource configuration (position) of i^{th} particle d^{th} dimension, R_{lbi} = local best resource configuration of i^{th} particle, R_{gb} = global best resource configuration, = velocity of i^{th} particle d^{th} dimension and V_{id}^{+} = new velocity of i^{th} particle d^{th} dimension. Figure 8.16 shows the process of particle movement in a PSO-driven DSE.

PSO is considered a very sensitive algorithm for applied optimization problems. Careful tuning of sensitivity parameters such as acceleration coefficient and inertia weight is very critical for the exploration process to attain faster convergence to optimal solution. However, the pivotal advantage of employing PSO for intractable multi-dimensional optimization problems is in the stochastic nature, which enables particles to reach every corner of design space (if needed) without losing the ability to maintain a fine balance between exploration-exploitation through linearly decreasing inertia weight. Mathematically it has been established in the literature [9], how proper tuning of sensitivity parameter values produces guaranteed convergence.

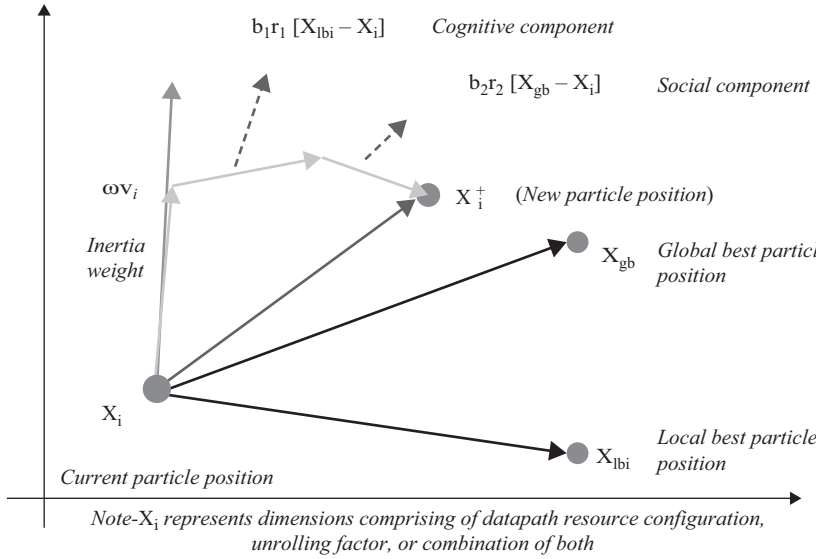
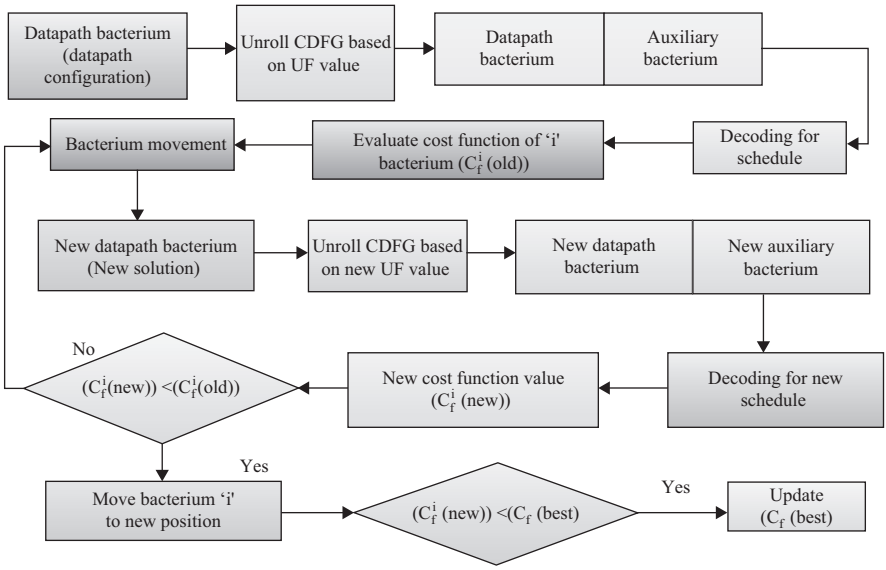


Figure 8.16 Particle movement in PSO-DSE framework

8.4.2 A BFOA-exploration process

The input to the framework is a CDFG of an application that describes the behavioral description of the datapath, set of user-specified design constraints for power and delay, and the module library consists of energy consumed by each resource in pico joule (Pj), latency of each resource in nanoseconds (ns), hardware area of each resource (number of transistor), and user-specified maximum availability of resources. For example assumed values are area of adder/subtractor = 2030 au, multiplier = 2464 au, and multiplexer = 126 au; delay of adder = 270 ns, multiplier = 11000 ns; where 1 au = 1 transistor; number/type of MUX/DMUX is directly extracted from the scheduling and pc = 29.33 nW. The flow chart of the proposed exploration process is shown in Figure 8.17 *(the corresponding pseudocode of the exploration process, ignoring details of encoding/decoding, is explained in Figure 8.18 later).* The method presented is based on a novel bacterium encoding scheme (comprising components called 'datapath bacterium' and 'auxiliary bacterium') for CDFG. This bacterium encoding scheme is accountable for exploration of optimal scheduling for CDFGs. The 'datapath bacterium' is capable of concurrently exploring the optimal resource configuration array and UF. On the contrary, the auxiliary bacterium is encoded by 'load value' and 'utilization value' metric which addresses precedence conflict between operations during scheduling. The 'auxiliary bacterium' acts as a support bacterium to its corresponding 'datapath bacterium' and is not subjected to evolutionary operation (or evolution) during exploration.



This process is repeated for each bacterium 'i' in the population (p) in each step (j) and then 'j' is incremented until termination under following condition:
1. When the algorithm reaches the maximum limit of the iteration.
2. When the C_f (best) does not change for 10 iterations.

Figure 8.17 Flowchart of the exploration methodology

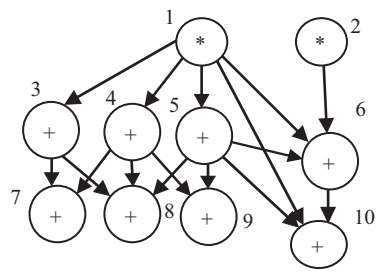


Figure 8.18 CDFG example

8.4.3 Encoding/initialization of the datapath bacterium

The presented datapath bacterium (Bn) of the proposed generic bacterium structure for CDFGs is provided in (8.1). This proposed datapath bacterium comprises two segments: (i) resource type array (i.e., array of resource configuration) and (ii) loop UF. The size of the initial population of the bacteria is arbitrarily assumed as three which are equally spaced in the design space. The first bacterium (B1) has been encoded with resource configuration which results in the serial implementation (indicating worst case latency among design solutions in the space). The second bacterium (B2) has been encoded with resource configuration which results in the maximum parallel

implementation (indicating best case latency among design solutions). The third bacterium (B3) is placed in the middle of the design space (midvalue (MV) between serial and parallel implementation bacteria). The initial configurations of the bacteria are described as follows:

$$\mathrm{Bn} = ((\mathrm{R1}), (\mathrm{R2}) \ldots (\mathrm{Rn}), (\mathrm{UF})) \tag{8.13}$$

$$\mathrm{B1} = (\min(\mathrm{R1}), \min(\mathrm{R2}) \ldots \min(\mathrm{Rn}), \min(\mathrm{UF})) \tag{8.14}$$

$$\mathrm{B2} = (\max(\mathrm{R1}), \max(\mathrm{R2}) \ldots \max(\mathrm{Rn}), \max(\mathrm{UF})) \tag{8.15}$$

$$\mathrm{B3} = (\ldots (\min(\mathrm{Rn}) + \max(\mathrm{Rn}))/2, (\min(\mathrm{UF}) + \max(\mathrm{UF}))/2) \tag{8.16}$$

Where R1,R2…,Rn are various resource types and UF is the loop UF.

8.4.4 Encoding of the auxiliary bacterium

Based on the initial population of datapath bacterium is created as described earlier, the unrolled untimed CDFG corresponding to the UF value (specified in the datapath bacterium) is generated for each parent. This indicates that for B1, B2, and B3 the corresponding unrolled CDFGs are generated for $\mathrm{UF}^{\min}$, $\mathrm{UF}^{\max}$, and $\mathrm{UF}^{\mathrm{MV}}$. An auxiliary bacterium is then constructed using an encoding technique corresponding to each unrolled untimed CDFG of bacterium population (B1, B2, and B3). This indicates that an auxiliary bacterium exists for each datapath bacterium. This auxiliary bacterium acts a priority resolver for operations competing during scheduling. The priority resolution (E) of opn 'oi' is performed through proposed encoding scheme given as:

$$\mathrm{E(oi)} = \mathrm{W1} * (\mathrm{LV}) + \mathrm{W2} * (\mathrm{UV}) \tag{8.17}$$

where LV is the load value and UV is the utilization value for each node (operation). W1 and W2 are designer-specified weights assumed as 0.5 each. The load value of an operation is the summed value of the load factor (delay) of each successor operations including the operation itself. The utilization value of an operation is its number of child branches. An example of encoding value of operations (for auxiliary bacterium) for an example CDFG (Figure 8.19) is provided in Table 8.1

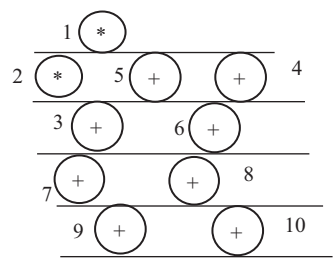


Figure 8.19 Scheduling with encoding

Table 8.1 Encoded values (E(OI)) for auxiliary bacterium for CDFG in Fig. 2

o1	*o2*	*o3*	*o4*	*o5*	*o6*	*o7*	*o8*	*o9*	*o10*
291.5	289.5	15	15.5	16	14.5	7	7	7	7

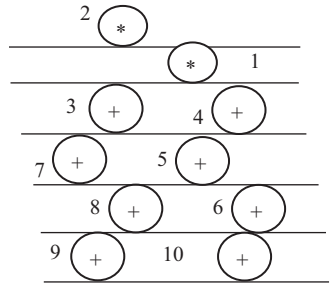


Figure 8.20 Scheduling with only load values

(*assuming: 1 mul: 550 control steps (CS), I adder: 14 CS*; *e.g., for opn 1 using (8.17), E (oi) = 0.5*(550 + 14 + 14) + 0.5*(5) = 291.5)*. Further, a motivational example on this encoding strategy, which integrates utilization value besides load value, is described below. *Scenario 1*: assume two encoded operations with load value (and having same load values), whereby the tie is broken by randomly selecting the operations during scheduling. *Scenario 2*: the two encoded operations have same load value. However, instead of randomly breaking, the tie is broken based on the proposed encoding strategy (function of load value and utilization value) during scheduling. Example, for a resource configuration, 2(+) and 1(*), the schedule in Figure 8.20 (for CDFG in Figure 8.19) uses only load values to resolve operation priority during scheduling and thereby consumes 6 control steps compared to schedule in Figure 8.21 which uses proposed encoding to resolve operation conflict during scheduling and consumes only 5 control steps.

8.4.5 Proposed movement of bacterium

Ignoring the details of bacterium encoding/decoding, the pseudocode of proposed exploration process is given in Figure 8.18. When a bacterium moves performs locomotion in every step (j), the proposed DSE mechanism explores new feasible solutions. However, after a designer-specified periodic intervals ('y^{th}' iteration step), the process of ED occurs. The ED algorithm is repeated for N_{ed} times (a temporary counter 'l' is initialized with the values of N_{ed} which decrements after every corresponding occurrence of ED operation; where 'N_{ed}' is a designer specified variable that defines the number for times, ED occurs. Therefore, it tracks the number of times ED is further allowed). Further, arrays (Ed [j–]) is created for ED process,

j (step/iteration) = 1
Repeat for 'p' *//p = bacterium population*
{
If j = (n*N_c/N_{ed}) // 1<= n <= N_{ed}
Then perform *Elimination-Dispersal mechanism*

Tumble: Generate a random vector Δ_m(i), within [–1,1].
Tumble-count=1
Swim-count=1
1. Generate a tumble vector Δ_m(i) .
2. $B_i^{new} = B_i^{current} + C(i) * (\frac{\Delta m(i)}{||\Delta m(i)||})$ (6)
//perform chemotaxis
3. Goto the cost function to evaluate the decoded CDFG and store it in C_f^i (new)
4. If $C_f^i(old) > C_f^i(new)$ and tumble-count<=5
Tumble-count = tumble-count+1.Gotostep1.
5. Else Swim-count < 5
$B_i^{current} = B_i^{new}$
Goto step 2.
}
j++
N_c is the maximum number of iterations/steps allowed
N_{ed} is the maximum number of elimination-dispersal steps allowed within N_c
B_i is the position of the i^{th} bacterium.
C_f^i(new)= new cost function value of 'i'th bacterium
C_f^i(old)= old cost function value of 'i'th bacterium

Figure 8.21 Pseudocode of exploration methodology

each to store the outcome, checking whether ED has been performed in last iterative step. This storage structures are necessary to determine whether variables 'y' needs up-gradation. If Ed [j–] has taken place, then, 'y' is updated.

1. *Chemotaxis mechanism*
The chemotactic function (8.10) incorporates a behavior of tumble/swim in order to explore the new positions; where C(i) is the step size taken in random direction specified by the tumble and Δ is a random vector whose elements lie in [−1, 1]. A substantially large value of C(i) is required for DSE, in order to avoid redundant solutions.

2. *Elimination-dispersal mechanism*
In order to implement the ED mechanism, new replacements are randomly initialized over the search space (between the least fit and best fit bacterium position but beyond their midpoint, but closer to the best fit bacterium) by eliminating the least fit bacterium. If the new replacement obtained is already found to be explored, and then dispersal is repeated. Further, if new cost of the dispersed bacterium position is

found higher than the replaced bacterium then it is not accepted. The ED mechanism has been adopted from Reference 6.

8.4.6 Models for metric

For evaluation of a bacterium (or candidate design solution), the following prediction models has been used from Reference 1.

1. *Execution time: prediction model*

$$T_E = \Delta * ((C_{first} + (UF\text{-}1) * C_{II}) * \alpha + (I \bmod UF) * C_{first}) \tag{8.18}$$

Where C_{first} represents CS count required to execute first iteration, I is the maximum count of iteration (loop count), and α is $\left(\frac{I}{UF}\right)^{quotient}$; Δ is the delay of one CS in nanoseconds; 'C_{II}' represents difference in CS count between outputs of consecutive iteration ns. Equation (8.18) is a prediction model for T_E, where the requirement of tediously unrolling the CDFG for a given UF is not needed to calculate T_E, unless # of independent operations required to be performed in parallel due to unrolling exceeds available resources (specified in bacterium) [1].

2. *Power*
The power models are described in (8.6–8.9).

3. *Cost function*
The fitness function (considering execution time/delay and power) of a solution from [1, 6] is defined as

$$C_f^{X_i} = \varphi_1 \frac{P_T - P_{cons}}{P_{max}} + \varphi_2 \frac{T_E - T_{cons}}{T_{max}} \tag{8.19}$$

where, $C_f^{X_i}$ = fitness of particle X_i; φ_1, φ_2 = user-specified weights for power and execution time; 'P_{cons}' and 'T_{cons}' are power and execution time constraints defined by the user; 'P_{max}' and 'T_{max}' are the normalized values of power and execution time of a potential design solution (bacterium).

8.4.7 Results of the BFOA-exploration process

The presented DSE approach has been implemented in Java and run on Intel core i5-2450 M processor, 2.5 GHz with 3 MB L3 cache memory and 4 GB DDR3 RAM. The results are presented in four phases: (i) sensitivity analysis: impact of varying population size on cost and convergence iteration (and exploration time) in Table 8.2. As evident in Table 8.2, for all benchmarks, as the population size increases, the convergence iteration decreases (however, the exploration time increases). This is because the computational complexity per iteration is more for larger population size. Nevertheless the cost of final solution (quality of results: QoR) remains same for all benchmarks regardless of population size. (ii) Results (solutions obtained) of

Table 8.2 Sensitivity analysis of the impact of population size on cost and convergence iteration

Benchmark [8, 14]	Population count	QoR	Exploration time (ms)	Convergence iteration
FIR	3	0.36	62	29
	5	0.36	81	26
	7	0.36	91	25
FFT	3	0.22	197	36
	5	0.22	253	35
	7	0.22	300	34
Differential equation	3	0.30	165	38
	5	0.30	184	37
	7	0.30	247	34
MPEG motion vector	3	0.29	47	26
	5	0.29	56	25
	7	0.29	68	24
JPEG down sample	3	0.65	63	25
	5	0.65	83	23
	7	0.65	109	23

Table 8.3 Results of proposed approach

Benchmark [8, 14]	Solution	Latency (us)		Power (mW)	
		Constraint	Proposed	Constraint	Proposed
FIR	UF = 4, 1(+), 4(*), 1(<)	60	24.16	0.50	0.45
FFT	UF = 2, 1(+), 1(−), 4(*), 1(<)	800	292.28	2.00	0.65
Differential equation	UF = 4, 1(+), 1(−), 4(*), 1(<)	600	267.24	1.20	0.63
MPEG motion vector	1(+), 5(*)	36	33.27	1.00	0.55
JPEG down sample	2(+), 1(*)	26	24.97	0.60	0.31

proposed approach for user constraints of power and delay is shown in Table 8.3. As evident in Table 8.3, the solution found indicates an optimal combination of resource array and loop UF CDFG's, where the final solution comprehensively meets the user constraints of power and delay (execution time) as well as minimizes the final cost (as per (8.5)). (iii) Results of comparison with existing approaches in References 3 and 5 in terms of QoR and exploration runtime. As evident in Table 8.4, the QoR of the approach is significantly better than that in References 3 and 5. This is because optimization of loop unrolling was not performed simultaneously with datapath resource configuration in them. Additionally, adaptive features such as tumbling which assists in changing direction when a search path is found ineffective does

Table 8.4 Results of comparison for QoR and exploration runtime

Benchmark [8, 14]	Final solution			Exploration run time			QoR (Cost)		
	Proposed	**[5]**	**[3]**	**Proposed (ms)**	**[5] (min)**	**[3] (s)**	**Proposed**	**[5]**	**[3]**
FIR	4(*), 1(+), 1(<), UF = 4	3(*), 1(+), 1(<),UF = 8	4(−), 1(+), 1(<), UF = 8	62	4.31	5.03	0.36	0.41	0.38
FFT	4(*), 1(+), 1(−), 1(<), UF = 2	3(*), 2(+), 1(−), 1(<), UF = 16	2(*), 1(+), 1(−), 1(<), UF = 16	197	>15	141	0.22	0.60	0.70
Differential equation	4(*), 1(+), 1(−), 1(<), UF = 4	4(*), 1(+), 2(−), 1(<), UF = 16	3(*), 1(+), 1(−), 1(<), UF = 16	165	>15	436	0.30	0.52	0.51
MPEG MMV	5(*), 1(+)	3(*), 1(+)	5(*), 1(+)	47	5.45	6.63	0.29	0.39	0.36
JPEG DS	1(*), 2(+)	1(*), 2(+)	1(*), 1(+)	63	2.5	8.21	0.65	0.65	0.77

Table 8.5 Results of quality metrics for proposed approach for quality metrics

Benchmark [8, 14]	Generational distance	MFE	Spacing	Spread	Weighted metric
FIR	0.00	0.48	0.17	1.01	0.51
FFT	0.00	0.75	0.03	0.80	0.40
Differential equation	0.00	0.55	0.02	0.88	0.44
MPEG MMV	0.00	0.07	0.07	0.95	0.48
JPEG DS	0.00	0.12	0.12	0.73	0.36

Table 8.6 Results of proposed approach with loop pipelining feature included

Benchmark [8, 14]	Solution	Latency (us)		Power (mW)		QoR (Cost)	Exploration runtime (ms)
		Constraint	Proposed	Constraint	Proposed		
FIR	UF = 4, 1(+), 4(*), 1(<)	60	24.16	0.50	0.45	0.36	94
FFT	UF = 2, 1(+), 1(−), 4(*), 1(<)	800	292.28	2.00	0.65	0.22	390
Differential equation	UF = 4, 1(+), 1(−), 6(*), 1(<)	600	224.32	1.20	0.75	0.31	353

not exist in genetic-based approaches. Besides above, the encoding of the solutions did not comprise utilization metric concept in determining priority during scheduling which helps in reducing latency. Therefore two different schedules (different latency) are possible for same resource configuration (this has been established before). The exploration runtime of [3] and [5] were higher because, both the previous approaches being driven through GA induces greater computational complexity than proposed fast bacterial foraging driven DSE process. (iv) Results for quality metric evaluation.

Table 8.5 gives the metrics: generational distance, maximum pareto-optimal front error, spacing, spread, and weighted metric which help in estimating the effectiveness of multi-objective optimization algorithms. The generational distance calculates the distance between the obtained non-dominated solutions and the true Pareto front. The value of '0' obtained in all benchmarks shows that all the non-dominated solutions found by the approach lie on the true Pareto front. Spread and spacing evaluate the diversity of the non-dominated solutions. The spacing is the relative distance between two consecutive non-dominated solutions, while the spread is a measure of their diversity with respect to the extremes of the pareto-optimal front. The weighted metric is a combined qualitative measure of both closeness and diversity. A low value of all these qualities, like the ones obtained in the table, mean that the proposed approach found a good, diverse set of non-dominated solutions. Table 8.6 indicates the result of the proposed approach with the inclusion of software-based loop pipelining feature

(the concept was described in Section previously). As evident from Table 8.6, for all the CDFGs except differential equation, the quality of solution remains same. However, as the loop pipelining concept mandates full unrolling, therefore, the maximum unroll factor (equals to the iteration count) which was usually discarded by the screening algorithm (described in Reference 15), is now required to be considered during exploration. This indicates that the exploration runtime is expected to increase as any solution with maximum unroll factor is also additionally evaluated during the exploration process. Therefore, as evident in Table 8.6, the exploration runtime of the CDFGs increases with incorporation of loop pipelining feature (however as mentioned with no change in final solution found).

8.5 HLS approaches for secure information processing

Secure information processing during HLS is a new subject of research in the CAD and VLSI community. As discussed earlier in the introduction, for secure and robust information processing during HLS design, detection of hardware Trojans (which are vindictive modifications in the logic of the circuit by an adversary) are very crucial [38]. Therefore, detection strategies for the SoC integrator for such malicious alterations have become thrust research areas recently [32–35]. Nonetheless, detection process of hardware Trojans requires supplementary hardware, which thereby may not abide the user constraints provided (if not accounted for during specification). The design process of hardware Trojan secured datapath should govern the practice of adaptive intelligent exploration approach based on user area-delay constraints. Further, a hardware Trojan secured datapath should be generated (on user constraints) with any quantity of hardware available. However during exploration, abiding the rule of assigning similar operations in original and duplicate units (for comparing) to distinct hardware (from different vendor) is necessary for detection. This creates an orthogonal condition to the requirement of designing a Trojan secured datapath with any quantity of hardware available (as well as satisfying user constraints) [38].

8.5.1 Related work

There have been almost no effort on hardware Trojan detection during HLS as well as generation of hardware Trojan secured datapath based user power-delay tradeoff during DSE. This involves exploring an optimized resource configuration as well as efficient vendor allocation procedure concurrently. For example, authors of References 35 and 36 adopted a concurrent error detection (CED) technique to only detect the malicious output during HLS using a diverse set of 3PIP vendors for a double modular redundant (DMR) system. However, the authors of References 35 and 36 did not include exploration of Trojan secured datapath based on user constraints of power and delay in their approach. Further, it did not perform detailed probing of the optimal distinct vendor allocation procedure (which affects final delay and power of the

schedule as will be established in the chapter later). Lack of deliberation of above led to results of an inferior quality final solution (higher cost). However, in Reference 36, the advantage is that it delivers retrieval from Trojan errors besides detection. In Reference 37, side channel analysis is performed and is capable of detecting malicious hardware alterations in the occurrence of large process variation induced noise. However, in Reference 37 no optimization is performed based on power-delay tradeoff during HLS. Overcoming the above limitations, an approach is presented in the next section, which generates an optimized hardware Trojan secured datapath using DMR logic based on user power-delay constraint during HLS.

8.5.2 Exploration process of hardware Trojan secured datapath: security against untrusted third party digital IPs

Simply the detection process of Trojan is not as upfront as CED of transient faults as it encompasses the concept of multiple 3PIP vendors to aid detection, let aside the exploration process of a user-optimized Trojan secured datapath based on multi-objective constraints. Vendor allocation procedure that is efficient needs to be formulated for Trojan detection during HLS, besides robust and adaptive exploration process for hardware Trojan secured datapath. *Note: Trojans that disable specific units/activities and produce no change in the output functionality computationally do not fall within the scope of this chapter*.

Figure 8.22 shows the method for DSE of optimal hardware Trojan secured datapath (resource configuration as well as optimal vendor allocation procedure) based on user power-delay constraint during HLS [38]. The exploration framework used is BFOA [6] which is well known for its stochastic nature as established previously.

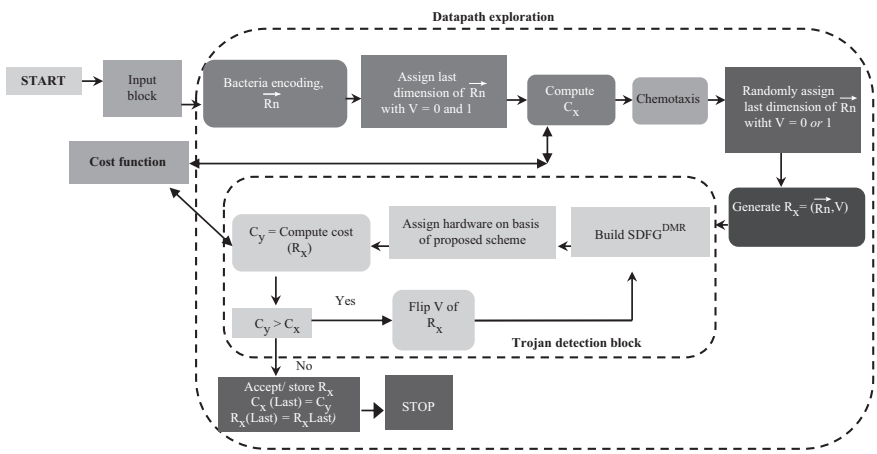


Figure 8.22 DSE methodology for generation of optimized hardware Trojan secured datapath [38]

Moreover, the exploration methodology is equally adaptable and efficient as PSO framework [1]. In Figure 8.22, the input block includes the module library, behavioral description of CDFG, and predefined user specified parametric constraints for power and time execution (or delay) to the exploration process. A set of control parameters such as 'N_c' (maximum number of chemotaxis steps allowed which is also one of the stopping criterion that indicates the maximum limit of the iterations that the exploration approach will execute) and 'p' are used for driving the BFOA based exploration process where 'p' specifies the number of individuals/bacterium (initial design solutions) actively playing a role in the evolutionary process [38]. The BFOA driven exploration process terminates for the following criteria: a) designer specified 'Nc' is reached; b) global best cost among bacteria population does not improve over last 10 iterations (chemotactic steps). Significant improvement in global best usually occurs after couple of iterations. *Note: Interested readers may refer to [6] for more details [38]*.

This BFOA-driven exploration process includes an encoding scheme for bacterium which comprises a resource array (resource configuration) and vendor allocation procedure type 'V'. Therefore, a bacterium position (design solution) is represented as x_i[38]:

$$X_i = (\overrightarrow{R_n}, V) \tag{8.20}$$

where $\overrightarrow{R_n}$ indicates the resource array (resource configuration, e.g., number of adders, multipliers). The reason behind incorporating the last dimension with vendor allocation procedure type 'V' is discussed in later sections.

The bacterium positions representing design solution and vendor allocation procedure, are used to generate a DMR schedule of the DFG ($SDFG^{DMR}$) using (6.10) described, with distinct vendor assignment rule to detect the presence of hardware Trojan. Since distinct vendor assignment can be achieved in various ways, hence, the optimal allocation procedure of hardware units from distinct vendor for similar operations (in original and duplicate) is explored through this presented scheme, to find an optimized scheduling (this is motivated in our next section) of a hardware Trojan secured datapath DMR system. The obtained Trojan secured DMR schedule is evaluated on metrics of power and delay to determine the cost of the Trojan-secured datapath solution. Power component due to functional resources, switching elements (MUX and DMUX), comparator as well as overhead incurred from internal buffering (temporary storage of operation output) all contribute to the cost of the Trojan secured datapath solution explored. Since in a DMR schedule, similar operations are being executed at different times in both copies (original and duplicate), hence internal buffering is required accounting for the cost [38]. The DMR system requires to keep the outputs from both units stored in some internal buffer to compare only when both outputs are ready. This process of evaluating design solutions (bacterium positions) evolves through BFOA-driven exploration process using chemotaxis mechanism to generate an optimal hardware Trojan fault secured DMR system that satisfies P_{cons}, L_{cons}, as well as minimizes hybrid cost [38].

8.5.2.1 Incorporating vendor allocation procedure 'V' in problem encoding: motivation of exploring this besides datapath resources

As established in Reference 35, for detection of hardware Trojan in 3PIP core, minimum of two vendors (providing similar/equivalent functional IP) are always needed to provide distinctness (*Note*: cases where Trojans lead to disabling hardware units/activities are ignored in this chapter, as it falls outside the scope of this work). As similar resource type/IP from two different vendors may have different area, power, and delay, therefore how two vendors are allocated inside a DMR scheduling (i.e., assignment process of each vendor IPs inside the DMR during allocation) dictates the final latency and power of entire DMR design. *Note*: It is assumed that multiplier and adder provided by vendor V1 has area = '2468 au' and '2034 au', latency = '10000 ns' and '265 ns', and energy = '10.0 pJ' and '0.80 pJ', while multiplier and adder provided by vendor V2 has area = '2464 au' and '2032 au', latency = '11000 ns' and '270 ns', and energy = '9.8 pJ' and '0.739', respectively [38]. Therefore, simply assigning distinct vendor to similar operations of a DMR schedule for detection without examining into the detailed procedure of which allocation (assignment) procedure of vendor type is better, may lead to missing an alternate better solution in context of DSE of an optimal Trojan secured datapath (*established in upcoming paragraph*). Therefore, an additional dimension, 'V' (indicating allocation procedure of IP's from different vendor type) which can either be '0' or '1' is incorporated in the bacterium encoding along with resource array for exploration of an optimal solution [38]. The value of 'V' as '0' or '1' is interpreted as follows [38]:

1. *Vendor allocation procedure (Type 1): V = 1*

All operations of a specific unit being strictly assigned to resources of **same** vendor type (say: all operations of original unit strictly assigned to **same** vendor 'V1' and all operations of duplication to **same** vendor 'V2').

2. *Vendor allocation procedure (Type 2): V = 0*

Alternate vendor assignment to operations in control step of a unit (example in Figure 8.21, operation 3 and 6, are assigned alternatively to 'V1' and 'V2'. Next multiplication if any would have been assigned to 'V1' alternately).

In both above cases, whenever there is a conflict of operation during scheduling between operation of U^{OG} and U^{DP}, preference is given to the operation of U^{OG} during scheduling. For a resource set $\overrightarrow{R_n} = 2(+), 5(*)$, there are two possible DMR schedules generated for IIR filter benchmark on the basis of V = 0 and 1, as seen in Figures 8.23 and 8.24. More specifically, for, $x_i = (2(+), 5(*), 0)$, the latency is 23,080 ns and power is 0.58 mW, while, for $R_x = (2(+), 5(*), 1)$, the latency is 22,080 ns and power is 0.88 mW. Clearly, a difference is observed in the delay and power of the two generated scheduling solutions both abiding by distinct vendor type assignment to similar operations for detect ability. Therefore, only using distinct vendor assignment without probing into the procedure of allocation of vendor type in DMR system may lead to missing of better alternative (or optimal) solution in context of DSE [38].

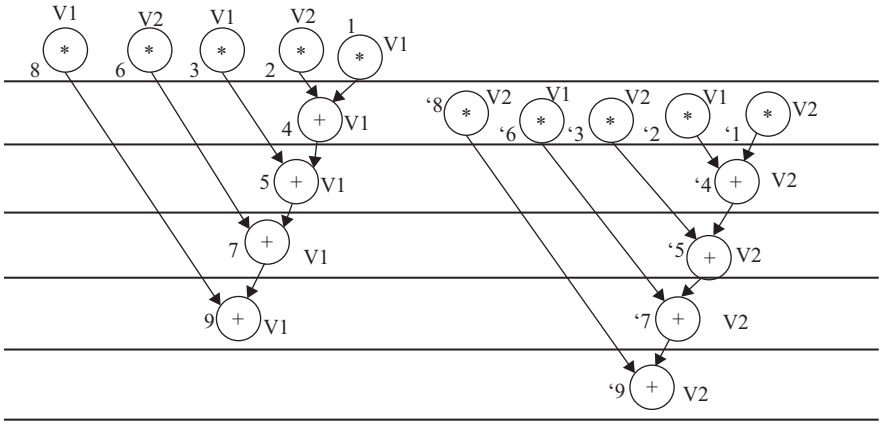


Figure 8.23 IIR filter for $V = 0$; $\overrightarrow{R_n} = 2(+), 5(*)$ indicating alternate assignment procedure of two vendor types [38]

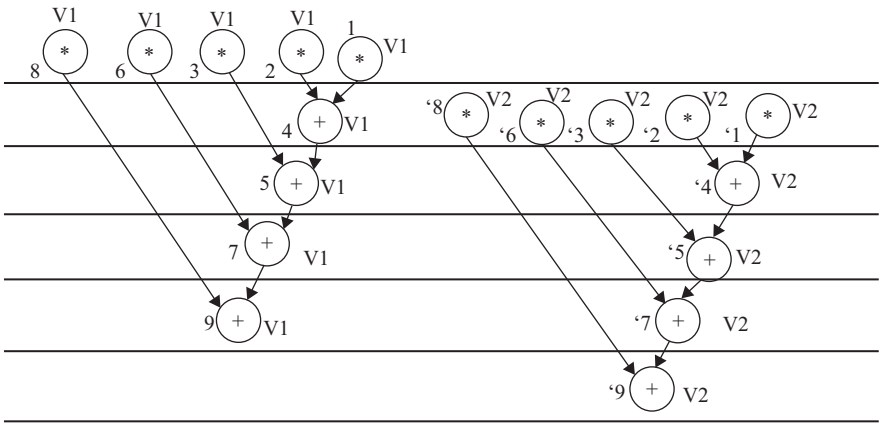


Figure 8.24 IIR filter for $V = 1$; $\overrightarrow{R_n} = 2(+), 5(*)$ indicating each entire unit strictly assigned to same vendor type (U^{OG} to ‘V1’ and U^{DP} to ‘V2’) [38]

8.5.2.2 Evaluation models

In this chapter, each bacterium position represents a resource array of datapath present in the design space as well as vendor allocation procedure [38].

a. *Power model*

Total power consumption (P_T^{DMR}) by a resource set is represented in terms of Static Power (P_S^{DMR}) and Dynamic Power (P_D^{DMR}). ‘P_T^{DMR}’ is represented as [38]

$$P_T^{DMR} = P_S^{DMR} + P_D^{DMR} \tag{8.21}$$

Static power (P_S^{DMR}) considered here is leakage power (due to leakage current) which is a function of resource area and leakage power per transistor [38].

$$P_S^{DMR} = \left(\sum_{j=1}^{2} \sum_{i=1}^{n} (A(R_i^{V_j}) * R_i^{Vj}) \right) * p_c \tag{8.22}$$

where the number of instances utilized from vendor V_j for a resource type R_i is represented as R_i^{Vj}, and the maximum number of resource type for vendor V_j is indicated by 'n', while $A(R_i^{Vj})$ is the area of a resource type (R_i) corresponding to vendor (V_j); 'p_C' is the power dissipated per transistor. Further, the average dynamic power as a function of dynamic activity of the resources of a resource configuration is given as [38]

$$P_D^{DMR} = \frac{E_{FU}}{L_T^{DMR}} \tag{8.23}$$

where E_{FU} is the total energy consumed by the resources. The power component includes power due to functional resources, interconnect units (MUX and DMUX), comparator (for error detection) as well as overhead incurred from internal buffering (during temporary storage of operation output in DMR scheduling). The model in (6.22) and (6.23) does not capture the effect of buffering and comparator, however, its impact was considered internally from scheduling during implementation.

b. *Delay (execution time) model*
For given 'D' functional resources the delay is [38]

$$L_T^{DMR} = \sum_{c.s=1}^{c.s(\max)} \sum_{j=1}^{2} \text{Max}(D(op_i^{Vj}), \ldots D(op_n^{Vj}), D(op_{'i}^{Vj}), \ldots D(op_{'n}^{Vj})) \tag{8.24}$$

where $1 \leq i \leq n$ and '$1 \leq$ '$i \leq$ 'n. (Here, operations in original and duplicate is labeled as i and 'i respectively; n and 'n = maximum number of operations in original and duplicate unit). Here, $D(op_i^{Vj})$ is the delay of operation i, assigned to vendor V_j, c.s represents control steps, while c.s(max) is the maximum number of control steps in a schedule.

c. *Cost model*
The proposed fitness function (considering total delay and power consumption of a solution) is defined as [38]

$$C_f(x_i) = W_1 \frac{P_T^{DMR} - P_{cons}}{P_{max}^{DMR}} + W_2 \frac{L_T^{DMR} - L_{cons}}{L_{max}^{DMR}} \tag{8.25}$$

where $C_f(x_i)$ is the cost of bacterium with resource set R_x, P_{max}^{DMR}, and L_{max}^{DMR} are the maximum power and delay of 0 the *DMR* system and W_1 and W_2 are the user defined weights both kept at ½ during exploration to provide equal preference; P_{cons} and L_{cons} are the user constraints for power and latency (delay).

Table 8.7 Results of approach [38]

Benchmark	Final solution for Trojan secured datapath [38]	Final solution for Trojan secured datapath [35]	Cost of final solution [38]	Cost of final solution [35]
DCT	4(+), 4(*), 0	5(+), 3(*),1	−0.106	−0.064
FIR	6(+), 6(*), 0	5(+), 5(*),1	−0.245	−0.209
ARF	2(+), 4(*),0	3(+), 3(*),1	−0.192	−0.056
MPEG MV	2(+), 10(*), 0	3(+), 8(*),1	−0.251	−0.226
IIR	2(+), 5(*), 0	2(+), 3(*),1	−0.125	−0.016
IDCT	6(+), 4(*), 0	5(+), 3(*),1	−0.154	−0.027

8.5.3 Results of exploration process of hardware Trojan secured datapath [38]

Approach in References 38 as well as 35 both have been implemented in java and run on Intel Core-i5-3210 M CPU with 3 MB L3 cache memory, 4 GB DDR3 primary memory, and processor frequency of 2.5 GHz. An average of 10 runs was reported for proposed BFOA DSE with equal weightage to both user objectives of power and delay ($W_1 = W_2 = 1/2$).

Table 8.7, illustrates the comparative results of the proposed approach and [35] when evaluated on the standard benchmarks. As seen from the results in Table 8.7, with the introduction of exploration for vendor allocation procedure type 'V' and user constraint driven exploration, the proposed approach generates better results in comparison to [35]. This is because, in previous approach there is no provision of exploring an optimal 'vendor allocation procedure' during scheduling in DMR as well as no optimization scheme based on user power-delay constraint for finding a better alternative solution. It may be noted that Collusion constraints have not been imposed for both approaches during implementation since preventing collusion that leads to disabling of hardware units falls outside the scope of this chapter. This chapter focuses on only hardware Trojan detection during DSE.

8.6 Selected tools available for HLS

HLS design process in emulated by EDA tools available in both commercial and open access form. These HLS tools are responsible for synthesizing the high level description of the application into its respective RTL circuit comprising of both datapath and control path. There have been many HLS tools proposed by both industries and academics, some of which are discussed in brief in this section. Various HLS tools with respect to availability and input format are shown in Figures 8.25 and 8.26, respectively.

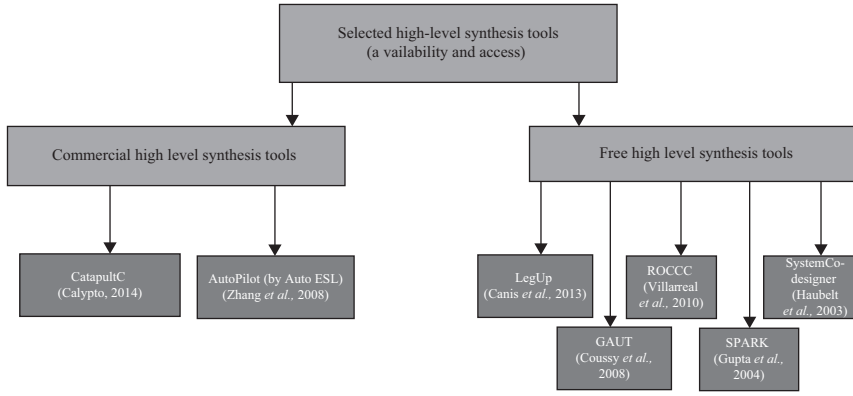


Figure 8.25 Various types of HLS tools classified in terms of availability

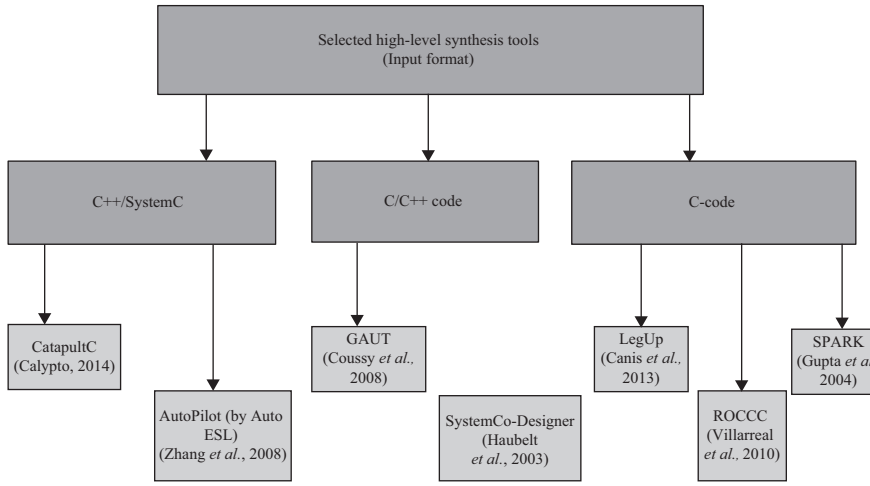


Figure 8.26 Different types of HLS tools classified in terms of input format

8.6.1 Selected commercial tools for HLS

Tools such as CatapultC from Mentor Graphics (now acquired by Calypto) [41] is a commercial EDA tool uses C/C++/systemC to describe the functional intent and generating RTL structure. The tool targets both Application Specific Integrated Circuit (ASIC) and field programmable gate arrays (FPGA) and works under the user constraints of time and area. This tool provides two options: (a) automated tool flow that disables the registers to make them inactive and (b) manual work flow that allows changing the code.

A tool called AutoPilot was introduced in Reference 42 to address the problem of exploration in HLS. It performs C/C++/systemC-to-RTL synthesis. The tool was targeted for FPGAs. It provides device specific verilog/Very High Speed

Integrated Circuit Hardware Description language (VHDL) RTL description. His tool is generally of interest to two class of users: (a) FPGA users who are aiming to improve the design productivity and (b) processor users who are looking to switch to FPGA to gain better performance and cost-performance ratio for computationally intensive applications.

8.6.2 Selected free HLS tools

Authors have proposed an open-source HLS tool called LegUp, which is for FPGA-based processor/accelerator systems. LegUp is able to synthesize C language to hardware, thereby providing a useful platform to perform HLS. Different FPGA architectures are supported by this tool, and allows of new scheduling algorithm and parallel accelerators. More specifically, LegUp is capable to compile the software program (written in 'C') into a hybrid architectural system comprising of FPGA-based soft core processor and hardware accelerator. Non-computation intensive tasks of the application which are not suitable for hardware implementation are executed in software of the processor. The tool also has the ability to explore hardware-software tradeoff of the design space [39].

Furthermore, tools such as ROCCC have also been proposed, which is an open-source HLS tool for generating RTL structure from C. It is designed for kernels that perform computation intensive tasks such as digital signal processing (DSP) applications. Therefore, ROCCC applies to a specific class of applications (streaming-oriented applications), and is not a general C-to-hardware compiler, unlike LegUp [39] which compiles larger C programs than is possible with ROCCC [40].

Moreover, there also exist some other free tools for HLS in the market such as GAUT [43] and SystemCoDesigner [44] have also caught attention in the EDA community such as GAUT which takes input a C/C++ description of the behavior description for automatically generating a RTL structure based on compulsory constraint of throughput (or initiation interval) and clock period. The output of the tool is the hardware structure generated comprising three major segments: processing unit, memory unit, and communication unit as well as the test bench to analyze the results.

The authors introduced a tool called SPARK for performing HLS. It accepts the C-code of the application which is synthesizable through SPARK tool to produce a VHDL output, that is further synthesizable by commercial logic synthesis tool. The tool targets FPGA which is heavily used for video and image processing applications owing to the reconfiguration capability of the FPGA. The tool considers the UF for the loop as being user-directed, thereby indicating that may not able to automatically determine the optimal combination of UF and datapath resource configuration simultaneously. SPARK incorporates various compiler and high-level transformation techniques to produce a scheduled, resource-bound datapath description. It is widely known in the academic community [7].

8.7 Conclusion and future directions of HLS

HLS of data intensive/control intensive application has been subject of major attention in CAD community for the last two decades. DSE process based on conflicting

objectives involved in HLS makes it non-trivial to solve for the researchers in this field. The exploration problem gets further convoluted with the inclusion of auxiliary variables of loop unrolling, loop pipelining, loop shifting, etc. Considering various loop manipulation techniques as well as datapath optimization based on user objectives during exploration of optimal scheduling is not inconsequential. This chapter besides providing a tutorial for HLS, also attempts to explain various concepts of loop manipulation, nature inspired DSE framework for solving the aforementioned problem as well as highlights some of the selected low power HLS methods. The chapter also introduced a novel DSE process using bacterial foraging optimization algorithm for solving the problem of finding a operation chaining based schedule during integrated datapath and loop manipulation (loop unrolling and loop pipelining) factor optimization. The exploration process works under the conflicting user constraints of power and delay imposed by the user. Additionally, models for estimating the delay of loop pipelined designs and loop unrolled designs were discussed. Further, the chapter describes some recent work on HLS approaches for secure information processing which involved untrusted third party IP cores. More specifically, detection of hardware Trojan in third party IPs during DSE of an optimized Trojan secured datapath based on user power-delay budget was elaborated in this chapter.

In terms of future research direction of HLS, nature inspired DSE frameworks may be leveraged to solve optimization problem of low power fault secured datapath of CDFGs that considers transient errors occurring due to single event transients. Additionally, factors such as linear energy transfer of particle during consideration of transient faults in HLS will be considered in the future by CAD community during development of exploration algorithms. Besides above, since optimization problem of fault tolerant datapaths for CDFGs based on power-delay budget is expected to gather significant interest, hence this area also lies at the center of our future investigation and development. In terms of security and trust in HLS, detecting hardware Trojans that ability to disable the activity of functional units is part of our future research in this area. Finally, the ongoing quest of improving the quality of final solution by including as much as lower level physical design details as possible during evaluation of design alternatives, is expected to be a research problem for future investigations for us and other CAD researchers.

References

[1] A. Sengupta, V.K. Mishra, 'Automated exploration of datapath and unrolling factor during power–performance tradeoff in architectural synthesis using multi-dimensional PSO algorithm', *Journal on Expert Systems*, 2014, vol. 41 (10), pp. 4691–4703.

[2] M. Holzer, B. Knerr, M. Rupp, 'Design space exploration with evolutionary multi-objective optimisation', *Proceedings of IEEE International Symposium on Industrial Embedded System*s, Lisbon, 2007, pp. 126–133.

[3] V. Krishnan, S. Katkoori, 'A genetic algorithm for the design space exploration of data paths using high-level synthesis', *IEEE Transactions on Evolutionary Computation*, 2006, vol. 10 (3), pp. 213–229.

[4] C. Mandal, P. P. Chakrabarti, S. Ghose, 'GABIND: A GA approach to allocation and binding for the high-level synthesis of data paths', *IEEE Transactions on VLSI Systems*, 2000, vol. 8 (5), pp. 747–750.

[5] A. Sengupta, R. Sedaghat, 'Integrated scheduling, allocation and binding in high level synthesis using multi structure genetic algorithm based design space exploration system', *Proceedings of 12th IEEE International Symposium on Quality Electronic Design (ISQED)*, California, 2011, pp. 486–494.

[6] A. Sengupta, S. Bhadauria, 'Automated exploration of datapath in high level synthesis using temperature dependent bacterial foraging optimisation algorithm', *Proceedings of 27th IEEE Canadian Conference on Electrical & Computer Engineering*, Toronto, 2014, pp. 1–5.

[7] S. Gupta, N. Dutt, R. Gupta, A. Nicolau, 'Loop shifting and compaction for the high-level synthesis of designs with complex control flow', *Proceedings of Design Automation and Test Europe (DATE)*, Paris, 2004, pp. 114–119.

[8] Express Benchmark Suite, *University of California Santa Barbara*, 2015, .http://express.ece.ucsb.edu/benchmark/.

[9] T. I. Cristian, 'The particle swarm optimisation algorithm: convergence analysis and parameter selection', *Information Process Letter,* 2003, vol. 85 (6), pp. 317–325.

[10] A. Mahapatra, C. S. Benjamin, 'Machine-learning based simulated annealer method for high level synthesis design space exploration', *Proceedings of IEEE Electronic System Level Synthesis Conference (ESLsyn)*, San Francisco, 2014, pp. 1–6.

[11] H. Y Liu, L. P. Carloni, 'On learning-based methods for design-space exploration with high-level synthesis', *Proceedings of 50th IEEE Annual Design Automation Conference (DAC)*, California, 2013, pp. 1–7.

[12] C. Schafer, K. Wakabayashi, 'Machine learning predictive modelling high-level synthesis design space exploration', *IET Computers Digital Techniques*, 2012, vol. 6 (3), pp.153–159.

[13] G. Xydis, V. P. Zaccaria, C. Silvano, 'A meta-model assisted coprocessor synthesis framework for compiler/architecture parameters customization'. *Proceedings of IEEE Design Automation and Test Europe (DATE)*, Grenoble, 2013, pp. 659–664.

[14] S. P. Mohanty, 'Energy and transient power minimization during behavioral synthesis', Ph.D. Dissertation, Department of Computer Science and Engineering, University of South Florida, FL, USA, 2003.

[15] A. Sengupta, V. K. Mishra, 'Swarm intelligence driven simultaneous adaptive exploration of datapath and loop unrolling factor during area-performance tradeoff', *Proceedings of 13th IEEE Computer Society Annual International Symposium on VLSI (ISVLSI)*, Florida, 2014, pp. 106–112.

[16] D. D. Gajski, N. Dutt, A. C. H. Wu, S. Y. L. Lin, *High Level Synthesis Introduction to Chip and System Design*, Kluwer Academic Publishers, USA, 1991, pp. 27–61.

[17] N. Raghunathan, S. Jha, Dey, *High-Level Power Analysis and Optimisation*, Springer, 1998, pp.175.

[18] M. Pedram, 'Power minimisation in IC design: principles and applications', *ACM Transactions on Design Automation of Electronic Systems,* 1996, vol. 1 (1), pp. 3–56.

[19] L. Benini, G. Micheli, 'System-level power optimisation: techniques and tools', *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 2000, vol. 5 (2), pp. 115–192.

[20] M. C. McFarland, 'The high-level synthesis of digital systems', *Proceedings of IEEE*, 1990, vol. 78 (2), pp. 301–318.

[21] M. C. McFarland, 'Tutorial on high-level synthesis', *Proceedings of 25th ACM/IEEE Design Automation Conference*, California, 1988, pp. 330–336.

[22] P. G. Paulin, J. P. Knight, 'Scheduling and binding algorithms for high-level synthesis', *Proceedings of 26th ACM/IEEE Design Automation Conference*, California, 1989, pp. 1–6.

[23] A. Raghunathan, N. K. Jha, 'Behavioral synthesis for low power', *Proceedings of IEEE International Conference on Computer Design*, 1994, pp. 318–322.

[24] N. K. S. Kurra, P. R. Panda, 'The impact of loop unrolling on controller delay in high level synthesis', *Proceedings of IEEE Design, Automation and Test Europe (DATE)*, California, 2007, pp. 391–396.

[25] L. Jirong, F. W. Weng, T. Mitra, 'A model for hardware realization of kernel loops'. In P. Y. K. Cheung, G. A. Constantinides (Eds), *LNCS,* Springer, 2003, vol. 2778, pp. 334–344.

[26] S. Gupta, 'Loop shifting and compaction for the high-level synthesis of designs with complex control flow', *Technical Report CECS-TR-03-14,* 2003, UC Irvine.

[27] R. K. Brayton, R. Camposano, G. Micheli, R. H. J. M. Gtten, J. van Eijndhoven, 'The Yorktown silicon compiler'. In D.D. Gajski, Ed. *Silicon Compilation*, Addison-Wesley, MA, 1988, pp. 204–311.

[28] Z. Peng, 'Synthesis of VLSI systems with the CAMAD design aid', *Proceedings of 23rd IEEE/ACM Design Automation Conference*, California, 1986, pp. 278–284.

[29] M. R. Rhinehart, J. Nestor, 'SALSA II: a fast transformational scheduler for high-level synthesis', *Proceedings of IEEE International Symposium on Circuits and Systems*, 1993, pp. 1678–1681.

[30] A. Sengupta, S. Bhadauria, 'Automated design space exploration of multi-cycle transient fault detectable datapath based on multi-objective user constraints for application specific computing', *Journal on Advances in Engineering Software,* 2015, vol. 82, pp. 14–24.

[31] J. Kennedy, R. C. Eberhart, 'Particle swarm optimisation', *Proceedings of IEEE International Conference on Neural Networks*, Australia, 1995, pp. 1942–1948.

[32] R. Karri, J. Rajendran, K. Rosenfeld, M. Tehranipoor, 'Trustworthy hardware: identifying and classifying hardware Trojans', *Proceedings of IEEE Computer*, 2010, vol. 43 (10), pp. 39–46.

[33] S. Bhunia, M. Abramovici, D. Agrawal, *et al*., 'Protection against hardware Trojan attacks: towards a comprehensive solution', *Proceedings of IEEE Design & Test*, 2013, vol. 99, pp. 6–17.

[34] X. Zhang, M. Tehranipoor, 'Case study: detecting hardware Trojans in third-party digital IP cores'. *Proceedings of IEEE International Symposium on Hardware-Oriented Security and Trust*, California, 2011, pp. 67–70.

[35] J. Rajendran, Z. Huan, O. Sinanoglu, R. Karri, 'High-level synthesis for security and trust'. *Proceedings of IEEE 19th International On-Line Testing Symposium*, Chania, 2013, pp. 232–233.

[36] C. Xiaotong, M. Kun, S. Liang, W. Kaijie, 'High-level synthesis for run-time hardware Trojan detection and recovery', *Proceedings of 51st ACM/IEEE Design Automation Conference (DAC)*, California, 2014, pp. 1–6.

[37] S. Narasimhan, D. Du, R Chakraborty, *et al*., 'Multiple-parameter side-channel analysis: a non invasive hardware Trojan detection approach', *Proceedings of 3rd IEEE International. Symposium on Hardware Oriented Security and Trust*, California, 2010, pp. 13–18.

[38] A. Sengupta, S. Bhadauria, 'Untrusted third party digital IP cores: power-delay trade-off driven exploration of hardware Trojan secured datapath during high level synthesis', *Proceedings of 25th IEEE/ACM Great Lake Symposium on VLSI (GLSVLSI)*, 2015, pp. 167–172.

[39] A. Canis, J. Choi, M. Aldham, *et al*., 'LegUp: an open-source high-level synthesis tool for FPGA-based processor/accelerator systems'. *ACM Transactions on Embedded Computer System*, 2013, vol. 13 (2), Article 24, 27 pages.

[40] J. Villarreal, A. Park, W. Najjar, R. Halstead, 'Designing modular hardware accelerators in C with ROCCC 2.0'. *Proceedings of IEEE International Symposium on Field-Programmable Custom Computing Machines*, 2010, pp. 127–134.

[41] Calypto, Calypto: http://calypto.com/en/products/catapult/optimize_for_power_with_hls#productinfo, 2014.

[42] Z. Zhang, Y. Fan, W. Jiang, G. Han, C. Yang, J. Cong, 'AutoPilot: a platform-based ESL synthesis system', *High-Level Synthesis*, Springer, New York, 2008, pp. 99–112.

[43] P. Coussy, C. Chavet, P. Bomel, 'GAUT: a high-level synthesis tool for DSP applications', *High-Level Synthesis: From Algorithm to Digital Circuits*, Springer, Netherlands, 2008.

[44] C. Haubelt, J. Teich, 'Accelerating design space exploration using pareto-front arithmetic's', *Proceedings of ACM/IEEE Asia and South Pacific Design Automation Conference*, Japan, 2003, pp. 525–531.

[45] S. P. Mohanty, *Nanoelectronic Mixed-Signal System Design*, McGraw-Hill, USA, 2015.

[46] A. Sengupta, R. Sedaghat, Z. Zeng, 'Multi objective efficient design space exploration and architectural synthesis of an application specific processor (ASP)', *Journal of Microprocessors and Microsystems,* 2011, vol. 35 (4), pp. 392–404.

[47] A. Sengupta, 'Rapid and efficient multi objective design space exploration in high level synthesis of computation intensive applications', Ph.D. Thesis, Ryerson University, Toronto, Canada, 2012.

[48] A. Sengupta, R. Sedaghat, Z. Zeng, 'A high level synthesis design flow with a novel approach for efficient design space exploration in case of multi parametric optimization objective', *Journal of Microelectronics Reliability,* 2010, vol. 50 (3), pp. 424–437.

[49] V. K. Mishra, A. Sengupta, 'Swarm inspired exploration of architecture and unrolling factors for nested loop based application in architectural synthesis', *IET/IEEE Electronics Letters*, 2015, vol. 51 (2), pp. 157–159.

[50] A. C. Williams, A. D. Brown, M. Zwolinski, 'Simultaneous optimisation of dynamic power, area and delay in behavioural synthesis', *IEEE Proceedings Computers and Digital Techniques*, 2002, vol. 147 (6), pp. 383–390.

[51] D. S Harish Ram. M. C. Bhuvaneswari, S. S. Prabhu, 'A novel framework for applying multiobjective GA and PSO based approaches for simultaneous area, delay, and power optimization in high level synthesis of datapaths', *VLSI Design Hindawi*, 2012, vol. 2012, Article ID 273276, 12 pages.

[52] P. Gianluca, C. Silvano, V. Zaccaria. 'Discrete particle swarm optimization for multi-objective design space exploration', *Proceedings of 11th IEEE EUROMICRO Digital System Design Architectures, Methods and Tools*, Parma, 2008, pp. 641–644.

[53] A. Sengupta, V. K. Mishra, 'Integrated particle swarm optimization (i-PSO): an adaptive design space exploration framework for power-performance tradeoff in architectural synthesis', *Proceedings of IEEE 15th International Symposium on Quality Electronic Design (ISQED 2014)*, California, 2014, pp.60–67.

[54] A. Sengupta, S. Bhadauria, 'Automated exploration of datapath in high level synthesis', *Proceedings of 27th IEEE Canadian Conference on Electrical and Computer Engineering*, Toronto, 2014, pp. 69–73.

[55] X. Xing, C. C. Jong, 'Floorplan-driven multivoltage high-level synthesis', *VLSI Design Hindawi*, 2009, vol. 2009, Article ID 156751, 10 pages.

[56] A. Sengupta, 'Design flow from algorithm to RTL using evolutionary exploration approach', *Application of Evolutionary Algorithms for Multi-Objective Optimization in VLSI and Embedded Systems*, Springer, 2014, pp. 113–123.

[57] P. Sarkar, A Sengupta, M.K Naskar, 'GA driven integrated exploration of loop unrolling factor and datapath for optimal scheduling of CDFGs during high level synthesis', *Proceedings of 28th IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, Halifax, May 2015, pp. 75–80.

Chapter 9

SPICEless RTL design optimization of nanoelectronic digital integrated circuits

Elias Kougianos[1] *and Saraju P. Mohanty*[1]

The previous chapter discussed various steps of high-level synthesis (HLS) which are used for design exploration of digital integrated circuits. It then discussed specific methods for dynamic power dissipation optimization as well as synthesis of hardware-trojan free digital integrated circuits. The methods relied on various bio-inspired algorithms for design space exploration. As complementary material of the previous chapter, this chapter presents HLS methods for leakage-optimal digital integrated circuit design exploration. Specifically, a paradigm shift approach is presented in which the complete HLS flow is performed without use of any electronic design automation (EDA) tool. All the associated tasks such as modeling, characterization, and optimization are performed using non-EDA tools, and hence this is called the "SPICEless" approach. For a specific objective of nanoelectronic digital integrated circuits, gate-leakage power dissipation is targeted.

9.1 Introduction

Application-specific circuits and systems for various requirements involving digital signal processing (DSP) are everywhere. DSP chips are part of media players, DVD players, bluray players, smart mobile phones, tablets, digital TVs, etc. These electronic systems have profound impact on society and are used continuously throughout the globe (refer to Figure 9.1). Application-specific circuits and systems are quite complex in terms of transistor count due to the need of high-throughput that involves a very large number of operations per unit time. Such application-specific circuits and systems (also referred as application-specific systems-on-chip) have stringent power budget to reduce energy consumption as well as specific needs for battery operated portable electronics [29, 30]. Complex digital integrated circuits are primarily fabricated using nanoscale complementary metal-oxide semiconductor (nano-CMOS) processes, a specific example of nanoelectronic technology. Nanoelectronic technology has made it possible to fabricate complex integrated circuit in limited silicon areas. However,

[1]University of North Texas, Denton, TX, USA

Figure 9.1 Representative instances of application-specific designs [29]

the use of nanoelectronic technology has made design iterations to achieve closure numerous as well as more effort intense as far as design engineers are concerned. The use of nanoelectronic technology also has changed the power dissipation components present in the overall power profile of individual devices and overall circuits or systems [31].

The design flows of complex digital circuits and systems use a divide and conquer approach in which the circuit or system is represented by various abstraction levels such as system level and architecture level. Design engineers work at a specific level of design abstraction using tools needed at that level and perform the design using the components present at a specific level. For example, at the architecture-level datapath components such as adders and multipliers can be used for design exploration of the application-specific integrated circuit. At this level, digital hardware description languages such as VHDL and SystemVerilog can be used [29, 30]. A specific example of automated design approach which is used at the architecture level is HLS [27, 30, 32, 35, 45]. The outcome of HLS is a register-transfer level (RTL) structure or a target architecture of the digital integrated circuit consisting of a datapath and controller. The three major steps of HLS consist of scheduling, allocation, and binding. Any optimization conducted during these HLS phases leads to an optimal RTL structure or RTL description. Depending on the datapath component library available, algorithms used for the HLS stages and constraints, different optimizations can be performed to obtain optimal RTL descriptions or architecture-level descriptions. Thus, HLS phase optimization, RTL optimization, and architecture-level optimization are used interchangeably. The previous chapter presented the basic steps of HLS, hence they will not be repeated again. This chapter presents such an optimization approach during the HLS phases to obtain gate-oxide leakage-optimal digital integrated circuits [32].

Design space exploration during HLS heavily relies on a datapath component library or RTL library of a specific technology node. The RTL library needs to be created by a design group dealing with design at the lower levels of design abstraction. The RTL library can also be purchased from a third party. Typically layout tools and SPICE simulations are involved in the design and characterization of such a RTL library. SPICE has been the cornerstone integrated circuit design and simulation tool [29, 48]. The design effort involved in such an RTL library creation is very high. It can be done only for the technology nodes whose processes are available. Hence it is not feasible for emerging or nanoelectronic technology based designs. Thus, a paradigm shift is presented in this chapter, in which layout design and SPICE characterization have not been used. This is called a "SPICEless" approach [19]. In the absence of foundry design rules for low-end nano-CMOS and nanoelectronic technologies, analytical models are derived to describe RTL component characteristics such as gate leakage and propagation delay considering various physical effects such as gate depletion, channel length modulation, mobility degradation, velocity saturation, and body-bias. These models are then used for RTL design exploration during HLS instead of an actual RTL component library. In this SPICEless approach, the HLS can progress without waiting for the availability of the RTL library.

As a specific example, the current chapter presents RTL optimization of gate-oxide leakage [32, 35, 45]. In a similar manner, optimization of other characteristics of digital integrated circuits such as dynamic power and subthreshold leakage can be considered. Leakage-optimal digital integrated circuits can have wide impact on mobile electronics. In general, the major sources of power dissipation in a nanometer CMOS circuit are capacitive switching, short-circuit current, static leakage, and gate-oxide tunneling. However, with the aggressive scaling of technology, the gate-oxide tunneling current (gate leakage) has emerged as a prominent component of power dissipation. For sub-65 nm CMOS technology where the gate-oxide (SiO_2) thickness is very small, direct tunneling (DT) current is the major form of gate-oxide leakage. In this chapter, we propose architectural-level analytical models to estimate gate leakage and then propose a methodology for its optimization during HLS. Since no foundry libraries are easily available for design and layout using technologies below 45 nm, the current chapter provides analytical models from first principles to calculate the gate leakage and the propagation delay of behavioral-level components considering various physical effects such as gate-depletion, channel length modulation, mobility degradation, velocity saturation, and body-bias. The current chapter then presents an algorithm for scheduling of datapath operations during HLS for automatic determination of the optimal gate-oxide thicknesses in a dual oxide thickness (dual-T_{ox}) approach, such that the overall gate leakage dissipation of a target datapath circuit is minimal. As the oxide thickness considered is very low, it may not remain constant during the course of fabrication, hence the algorithm takes process variation into consideration. Behavioral components for both 65 nm and 45 nm CMOS technologies are characterized in order to study the trends of gate leakage as technology scales and provide them as inputs to the algorithm. The current chapter also presents extensive experiments for several behavioral synthesis benchmarks under various constraints to prove the effectiveness of the proposed approach. It is observed that gate-oxide

Table 9.1 Notations and parameters used for modeling and calculations

Parameters/ notation	Description with units
V_{dd}	Supply voltage in V
V_{gs}	Gate-to-source voltage in V
V_{Th}	Threshold voltage in V
V_{fb}	Flat-band voltage in V
V_{ox}	Voltage across the gate dielectric in V
V_{poly}	Voltage across polysilicon in V
V_{bs}	Body-to-source voltage in V
V_{dsSat}	Saturation drain voltage in V
I_{ox}	Gate-oxide leakage current in A
P_{ox}	Gate-oxide leakage power in W
I_{DSat}	Saturation drain current in A
ϕ_B	Barrier height for the gate dielectric in eV
ψ_S	Surface potential in V
C_L	Output load capacitance in F
C_{ox}	Gate capacitance in $\frac{F}{m^2}$
Q_B	Depletion charge density in $\frac{Coulomb}{m^2}$
μ_{sub}, μ_0	Bulk mobility in $\frac{cm^2}{V-s}$
θ	Mobility degradation factor per V
v_{sat}	Electron saturation velocity in $\frac{cm}{s}$
v_{norm}	Proportionality constant in $\frac{cm}{s}$
α	Physical constant modeling carrier saturation velocity
α_{sw}	Switching activity probability
$N_{channel}$	Channel doping concentration per cm^3
N_{poly}	Polysilicon gate doping concentrations per cm^3
N_{sub}	Substrate doping concentration per cm^3
n_i	Intrinsic concentration per cm^3
T_{ox}	Electrical equivalent oxide thickness in nm
L	Channel length of MOSFET in nm
W	Width of MOSFET in nm
ε_{ox}	Permittivity of gate dielectric in $\frac{F}{m}$
ε_{Si}	Permittivity of Si in $\frac{F}{m}$
q	Electronic charge in $Coulomb$
$h, \hbar$	Planck's constant in $J-s$
T	Temperature in degrees Kelvin (K)
k	Boltzmann's constant in $\frac{J}{K}$
m_o	Rest mass of electron in Kg
k_m	Constant for mass calculation; 0.19 for electron and 0.55 for hole
m_{eff}	Effective mass in Kg
η	Subthreshold slope factor
T_T	Transition time in s
T_{pd}	Propagation delay in s
I_{ox}NAND	Average gate leakage of a NAND logic gate
I_{ox}FU	Average gate leakage of a functional unit or datapath resource
T_{pd}FU	Propagation delay of a functional unit or datapath resource
$P_{ox\,FU}(c, r)$	Average gate leakage of the rth functional unit active in the control step c
$T_{pd\,FU}(c, r)$	Propagation delay of the rth functional unit active in the control step c
ST	Denotes single oxide thickness traditional case
DT	Denotes dual oxide thickness case
ΔP_{ox}	Percentage reduction in gate-oxide leakage power
ΔT_{pd}	Percentage penalty in critical path delay
N_c	Number of control steps
$n_{FU\,c}$	Number of resources active in any control step c

leakage reduction as high as 87.8% (on an average of 75.3%) for 65 nm and a maximum of 75.3% (on an average of 64.5%) for 45 nm process technology nodes can be achieved. It is also observed that for both technologies the average time penalty is approximately 19%. The notations and symbols used in the current chapter are listed in Table 9.1.

The remaining of the chapter is organized in the following manner: The big picture of SPICEless RTL optimization during HLS is presented in Section 9.2. The issue of power dissipation in nano-CMOS circuits is presented in Section 9.3. Various existing related research works are summarized in Section 9.4. The HLS methodology for minimizing gate-oxide leakage is discussed in Section 9.5. The SPICEless characterization methodology that generated analytical models for RTL components is presented in Section 9.6. The specific experimental results validating the proposed methodology are discussed in Section 9.7. The findings and conclusions of the proposed research along with suggestions for future research are presented in Section 9.8.

9.2 The concept of SPICEless RTL optimization during HLS

In order to make HLS suitable for nano-CMOS integrated circuits, the objective is to develop models that capture gate-oxide leakage and optimize during the HLS phases. The behavioral level is not as highly abstracted as system level nor as lowly abstracted as gate or transistor level. Hence, at behavioral level, there is a balanced degree of freedom to explore power reduction mechanisms, and it can help in investigating lower power design alternatives prior to circuit layout in actual silicon [31, 58]. In this section, the proposed SPICEless RTL optimization during HLS is presented in comparison with the classic RTL optimization during traditional HLS. The traditional RTL optimization during HLS flow is presented in Figure 9.2(a) [30, 50]. The basic steps of this flow have been discussed in reasonable detail in the previous chapter.

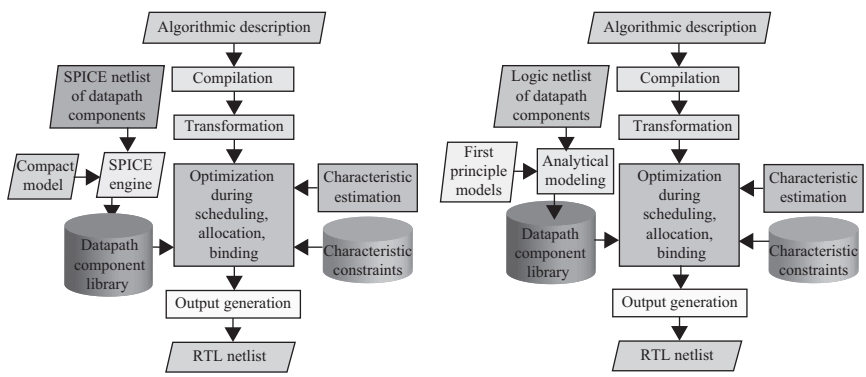


Figure 9.2 The HLS flows with SPICE and without SPICE. (a) HLS flow with SPICE and (b) SPICEless HLS flow

The steps are briefly revisited in this section as a matter of completeness of this chapter. The compilation step translates the input behavioral description in the form of SystemVerilog or VHDL to a data flow graph (DFG) data structure representation. The steps of HLS include loop transformation, loop unrolling, etc. which are operations over the DFG of the behavioral representation of the input algorithm for which a hardware design is to be performed. The RTL optimization for various objectives and constraints can be performed during the scheduling, allocation, and binding phases of HLS. In this HLS flow, the datapath component library or RTL library is used for design exploration through optimization during scheduling, allocation, or binding. The RTL library in a typical case is created and characterized through layout design and SPICE simulations. The RTL optimization assumes various constraints based on the target application of the design. During the RTL optimization the characteristic estimation is performed over the DFG of the target integrated circuit. The output generation step of HLS creates the datapath and control of the architecture using SystemVerilog or VHDL.

The proposed SPICEless HLS flow is presented in Figure 9.2(b) [32, 35, 45]. In this flow, the basic steps of HLS for RTL optimization are the same as the traditional steps such as compilation, transformation, scheduling, allocation, binding, and output generation. The SPICE characterization step which needs significant engineering effort has been replaced by a SPICEless characterization step. Many alternative options of SPICEless modeling and characterization are possible such as first principle physics based modeling and Simulink® or Simscape® based simulations [19, 29, 32]. The first principle physics based models can be implemented in high-level languages such as C or MATLAB®. Then using the first principle models the RTL components can be modeled in the form of analytical models and characterized for use. An alternative method is to model the RTL components using the primitives of Simulink® or Simscape®, even in the domain-specific languages supported by them [19], if desired. The RTL components can then be simulated and characterized in the Simulink® or Simscape® frameworks. The HLS phase optimizations can be performed in these high-level languages and frameworks using the large selection of available algorithms in MATLAB® and similar frameworks. In this HLS flow, all the steps can be performed outside EDA without using any EDA tools. Thus the design flow can be quite fast through fast simulation, characterization, and design exploration. The complete HLS flow can be a fully unified tool and framework. Most importantly, the SPICEless HLS flow is not dependent on any compact models, physical design, and hence can be used for current nanoelectronic as well as emerging technology based digital designs.

9.3 The issues in RTL optimization of power dissipation in digital circuits

Nanoelectronic technology based design can have many issues such as power, leakage, performance, reliability, etc. [29]. Design for X is design for excellence (DFX or DfX) in which X denotes the set of objectives corresponding to these issues, for example,

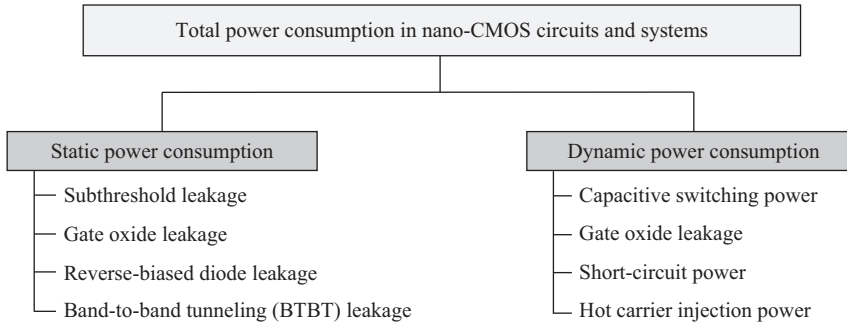


Figure 9.3 Various components of power dissipation in classic nano-CMOS based circuits and systems

design for power for power dissipation. Several issues such as battery life, reliability, thermal considerations, and environmental concerns have driven the need for low power designs. Power dissipation or energy dissipation has been and still remains a critical issue due to the fast proliferation of mobile computing such as tablets and smart phones. In the current chapter, power dissipation has been used as a case study for the proposed HLS flow. In particular, leakage power dissipation is considered as it is critical to reduce the standby power dissipation of smart phone like systems which have large leakage power dissipation and heavily depend on battery as the energy source.

The decreasing feature size due to aggressive technology scaling along with high packing density and the increasing clock frequency result in high on-chip electric fields which has made reliability a significant challenge for designers [38, 62, 66]. With aggressive technology scaling, both static and dynamic power have become equally contributing factors for the total power dissipation of a CMOS circuit [22, 66]. The various forms of power dissipation in a nano-CMOS based digital integrated circuit are presented in Figure 9.3 [5, 8, 29, 35, 46]. In a short-channel nanometer transistor, several forms of leakage current exist, such as reverse biased diode leakage, subthreshold leakage, gate tunneling current, hot carrier gate current, gate induced drain leakage (GIDL), and channel punch through current [55]. Of all these leakage mechanisms, SiO_2 tunneling current that flows during both active and sleep modes of the circuit is a significant component for low-end nano-CMOS technology (i.e., sub-65 *nm*). This is because low-end nano-CMOS uses ultra-thin oxide and is susceptible to new leakage mechanisms due to tunneling through gate oxide, which leads to gate-oxide current (I_{ox}) [22, 35, 65]. Thus, the major sources of power dissipation in a nano-CMOS circuit can be summarized as capacitive-switching power (P_{sw}), short-circuit power (P_{sc}), static power (P_{static}), and gate-oxide leakage power (P_{ox}) [5, 8, 29, 35, 46]:

$$P_{total} = P_{sw} + P_{short} + P_{static} + P_{ox}, \tag{9.1}$$

$$= \alpha_{sw} C_{sw} {V_{dd}}^2 f_{clk} + \tau \alpha V_{dd} I_{sc} f_{clk} + V_{dd} I_{leak} + V_{dd} I_{ox}. \tag{9.2}$$

In the above expressions, α_{sw} is the switching activity, C_{sw} is the total capacitance seen at the gate output, V_{dd} is the supply voltage, f_{clk} is the operating frequency, τ is the time for which short-circuit occurs, I_{sc} is the short-circuit current, I_{leak} is the leakage current, and I_{ox} is the average gate-oxide tunneling current. In the current chapter, the focus is on the reduction of gate-oxide leakage power of nano-CMOS datapath circuits during behavioral synthesis, HLS, architecture-level synthesis, or algorithm-level synthesis.

For the optimization of leakage power, the primary contribution of the current chapter is the demonstration of a dual-T_{ox} approach for reduction of gate-oxide leakage HLS using a SPICEless or non-EDA flow. The contributions of this chapter are of multiple forms. First, analytical models for gate-oxide leakage and propagation delay are presented for the characterization of functional units or datapath component libraries. The component library containing various architecture-level resources is characterized for different gate-oxide thicknesses to be used in a dual-T_{ox} technology. Subsequently, it is assumed that such functional units are made available as standard cells. The current chapter introduces an algorithm for scheduling of the datapath operations such that overall gate-oxide leakage of a datapath integrated circuit is minimal under given timing or resource constraints. It is assumed that all transistors used in a functional unit or architecture-level resource (such as adder and multiplier) have oxide of equal thickness, but the thicknesses of different functional units may be different. The functional unit using higher oxide thickness transistors dissipates less gate-oxide leakage power, but has larger delay. Such a functional unit can be used in the off-critical path of a circuit, to achieve the conflicting objective of power reduction and maintaining performance. On the other hand, a functional unit which uses lower oxide thickness transistors exhibits less delay and is suitable to be utilized in the critical path of a circuit. As the oxide thickness we are dealing with is very low, it may not remain constant during fabrication, hence the proposed algorithm takes process variation into consideration on the fly using the analytical models.

9.4 Power optimization at RTL: state-of-the-art

This section discusses the state-of-the-art of the techniques proposed for power optimization at the RTL [29, 30, 31]. The techniques focused on are the ones that use some form of optimization during HLS phases. The section then presents the concept of dual-T_{ox} approach for RTL leakage optimization for the advancement of state-of-the-art.

9.4.1 Existing methods for RTL power optimization

Power reduction in digital integrated circuits in general can be achieved at various levels of design abstraction, such as system, architectural, logic, transistor, and physical level. At each level of design abstraction, researchers have proposed different techniques for reduction of various sources of power dissipation. An overview of various techniques available for power optimization at the architecture or RTL is

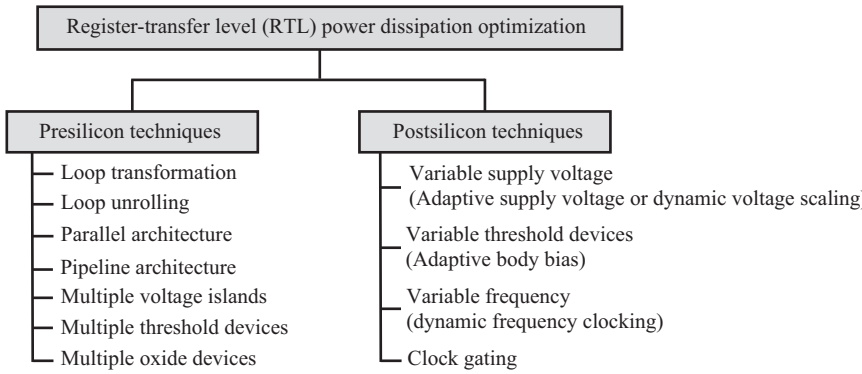


Figure 9.4 Various techniques for RTL power optimization

given in Figure 9.4 [29, 30, 31]. A method can be used for one or more forms of power dissipation. At the same time, several methods exist for reduction of a specific form of power dissipation. Pre-silicon techniques are developed and implemented during the design phase of the integrated circuits. The pre-silicon method then translates in the integrated circuit after its fabrication and cannot be tuned or modified afterwards. Thus, these methods can be considered as static methods. Pre-silicon methods include technology-independent techniques such as loop transformations and technology-dependent techniques such as multiple supply voltages and multiple threshold voltages. On the other hand, post-silicon techniques are incorporated during the design phase, but tuned or controlled after the integrated circuits are manufactured. Thus, these methods can be considered as dynamic methods. Post-silicon methods include techniques such as adaptive supply voltage mechanisms and adaptive threshold voltage mechanisms.

Dynamic power management (DPM) techniques, dynamic voltage (frequency) scaling (DVS), and clock gating are popular system level methods [4, 61]. The voltage scaling techniques such as multiple supply voltages (static mechanism) or variable supply voltage (dynamic mechanism) have been well researched for switching activity or dynamic power reduction in the last decade [4, 30, 31]. Several mature techniques have resulted out of these research works and have been deployed in real-life designs. Various multiple supply voltage (multi-V_{dd}) techniques have been explored as an attractive method for behavioral and RTL dynamic power minimization [2, 26, 38, 39, 59]. As a complementary mechanism, an associated technique called dynamic frequency clocking or dynamic frequency scaling is also explored during HLS [25, 27]. Voltage and frequency scaling together can help to reduce power consumption as well as energy consumption more effectively. The existing approaches explore reduction of various forms of power dissipation including average power, peak power, or power fluctuation for specific characteristics of the target application [40, 41, 75].

Dynamic power consumption, with the major component being switching activity power, has received a lot of attention and has been addressed. At the low-end nano-CMOS technology, gate-oxide leakage has been an issue [29, 31, 60].

A high-performance CMOS device will require gate-oxide thicknesses of 0.7–1.2 nm, thus making it more susceptible to new leakage mechanisms due to tunneling through gate oxide leading to gate-oxide current. The multiple oxide thickness, i.e., transistor gate-oxide thickness or specifically dual-T_{ox} method is proposed as a method for gate-oxide leakage optimization [35, 63, 64, 65]. The invention of high-κ based technologies eliminated the issue of gate-oxide leakage [6, 33]. A technique complementary to dual-T_{ox} called dual-κ has been explored for RTL optimization [43]. A firefly algorithm based RTL optimization approach that minimized the gate-oxide leakage and propagation delay of integrated circuit for various resource constraints is available [23]. However, design houses that do not have access to the cutting-edge high-κ/metal-gate processes still find gate-oxide leakage as a major issue.

There are several methods available for reducing sleep mode leakage, such as multi-V_{Th} [18, 21, 51, 53, 54], body-biasing [49], and state assignment [24]. In References 20 and 21, the dual-V_{Th} technique has been proposed for subthreshold leakage analysis and reduction during behavioral synthesis. The algorithms target the least used modules as the candidates for leakage optimization. In References 17 and 18, a multi-V_{Th} approach is used for reduction of subthreshold current during HLS. Binding algorithms have been proposed for power, delay, and area trade-offs. While a clique partitioning approach is used in Reference 18, a Knapsack based binding algorithm is proposed in Reference 17. The value of threshold voltage of short-channel transistors which in turn affect subthreshold leakage is dependent on various process and device parameters including doping concentration, gate-oxide thickness, silicon and gate-oxide permittivity, electronic charge, body-bias voltage, drain voltage, drain-induced barrier lowering (DIBL), and channel length [72, 36]. Adaptive body bias is a dynamic approach in which body-bias voltage (V_{bs}) is dynamically adjusted to change the threshold voltage (V_{Th}) which in turn controls leakage [70]. Multiple gate transistors (also known as 3D transistors) as well as FinFETs have been developed for reducing standby leakage as well as faster switching of the devices [16].

9.4.2 Multiple oxide thickness technology for gate-oxide leakage optimization

Gate-oxide leakage can be due to either DT or Fowler–Nordheim (FN) tunneling mechanisms which differ in the form of the potential barrier [32, 55]. The probability of carrier tunneling is a strong function of the barrier height (i.e., the voltage drop across gate oxide) and the barrier thickness. So, determining options for reduction of active leakage power dissipation is a requirement for new technologies. For supply voltage V_{dd}, effective gate-oxide thickness T_{ox}, the tunneling current dissipation in a CMOS can be described as [10, 22, 35],

$$I_{ox} = \beta W_{gate} \left(\frac{V_{dd}}{T_{ox}} \right)^2 \exp\left(-\gamma \frac{T_{ox}}{V_{dd}} \right), \tag{9.3}$$

where β and γ are experimentally derived factors. From the above equation, it is observed that the following possible options are available for reduction of gate leakage

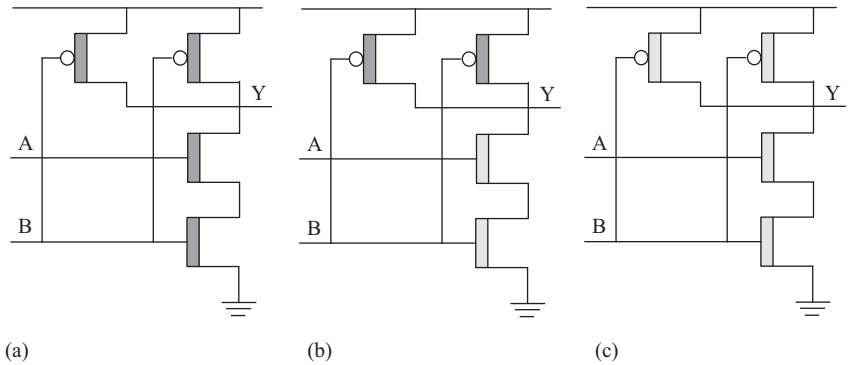


Figure 9.5 Nano-CMOS logic circuits to be used in a dual oxide technology for leakage power optimization. (a) With all nominal-T_{ox} devices, (b) high-T_{ox} NMOS devices, and (c) with all high-T_{ox} devices

power consumption, (i) decreasing supply voltage (V_{dd}), (ii) increasing gate-oxide thickness (T_{ox}), and (iii) decreasing gate width (W_{gate}). Decreasing power supply voltage is used as a popular option to reduce dynamic power consumption, and it will play its role in the reduction of leakage power as well. Increase in the gate SiO_2 thickness leads to an increase in propagation delay. Moreover, reduction of gate width may not be an attractive option as gate leakage current is only linearly dependent on it. Thus, it can be concluded that use of multiple gate-oxide thicknesses can serve as a leakage power (current) and performance trade-off. In this chapter, we explore the multiple thickness (multi-T_{ox}) approach for reduction of DT gate current during behavioral synthesis. It may be noted that the proposed multi-T_{ox} technique can be used along with any of the other available techniques, such as multi-V_{dd}, multi-V_{Th} or clock gating to provide a complete low power solution for sub-65 nm CMOS technology digital integrated circuits.

In dual-T_{ox} technology, nominal-T_{ox} devices, logic gates, or RTL components are selectively replaced with corresponding high-T_{ox} components for gate-oxide leakage reduction while maintaining the target performance. The idea of dual-T_{ox} technology is illustrated in Figure 9.5 [31, 43, 45]. Figure 9.5(a) shows a nominal logic gate with all nominal-T_{ox} devices. In Figure 9.5(b), the leaky NMOS devices are constructed with high-T_{ox} oxide. This is more close to the well-established dual-V_{Th} technology. In Figure 9.5(c), the logic gate is made of all high-T_{ox} devices. In this chapter, the use of a mix of RTL components of type (a) and type (c) can serve as gate-oxide leakage and performance trade-offs and will go well with industry trends. During HLS, selection of high-T_{ox} and nominal-T_{ox} (also called as low-T_{ox}) RTL components is performed for trade-off analysis. The key research questions include the following:

(1) How to identify how many of high-T_{ox} and low-T_{ox} resources to use for design trade-offs?
(2) How to select a mix of these resources for trade-offs?

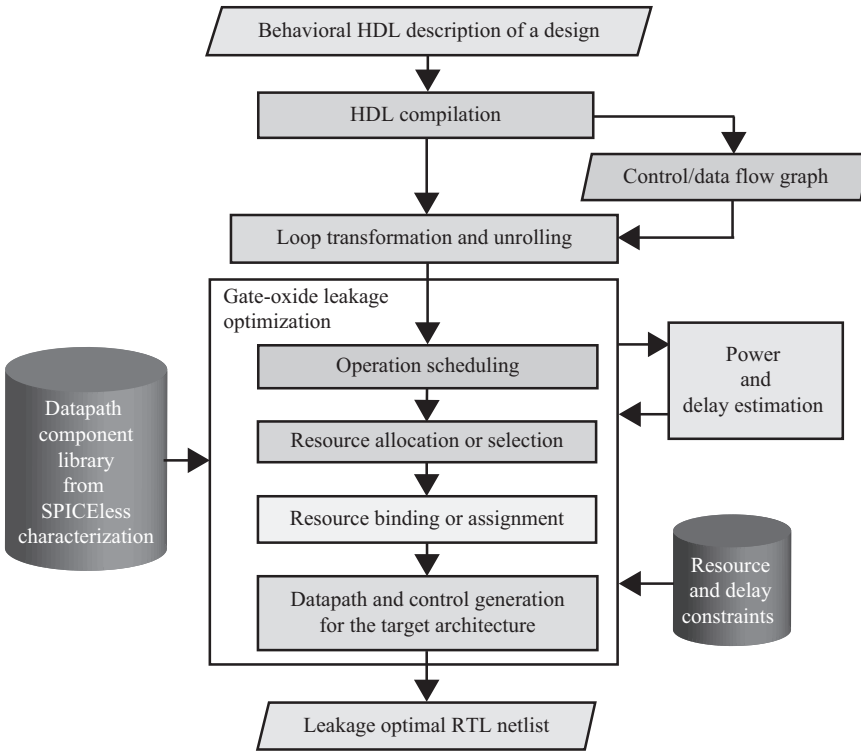

Behavioral HDL description of a design
HDL compilation
Control/data flow graph
Loop transformation and unrolling
Gate-oxide leakage optimization
Operation scheduling
Resource allocation or selection
Resource binding or assignment
Datapath and control generation for the target architecture
Power and delay estimation
Datapath component library from SPICEless characterization
Resource and delay constraints
Leakage optimal RTL netlist

Figure 9.6 HLS flow for gate-oxide leakage-optimal RTL generation

(3) How to schedule different operations to these resources for specified time constraints?
(4) How to bind these operations to the different resources for trade-offs?
(5) How to perform leakage, timing, and silicon area trade-offs?

9.5 A specific SPICEless RTL optimization approach

This section discusses a specific SPICEless RTL optimization flow during HLS. First, the overall flow is presented to give a broad picture of the optimization. Then a detailed discussion of the optimization objective is presented. Finally, a heuristic algorithm is presented for design space exploration during HLS.

9.5.1 The overall RTL optimization flow

An overview of the proposed SPICEless optimization flow during HLS is presented in Figure 9.6. As shown, there are several steps involved in the HLS, such as compilation, transformation, datapath scheduling, functional unit allocation, operation binding,

connection allocation, and architecture (consisting of datapath and control) generation. During the compilation and transformation phases, behavioral VHDL (or SystemVerilog) is compiled to structural VHDL (or SystemVerilog) to obtain a DFG or control/DFG. Resource or time constrained schedulers time stamp the variables and operations in the DFG so that the operations in the same group can be executed concurrently. While allocation fixes the number and types of resources to be used in the datapath circuit, the binding process involves attaching operations to functional units and variables to memory units. The connection allocation step determines the types or number of buses, buffers, and multiplexors for the communication between resources. Finally, the datapath and control of the target architecture are generated. Scheduling and binding are the major phases of low-power HLS, and these steps are explored for use for leakage, delay, and area trade-offs. The HLS phases are followed by logic synthesis and physical synthesis to generate the layout of the integrated circuit under design. The logic synthesis and physical synthesis phases of the digital integrated circuit design flow have not been shown in this figure as these are out of scope of this chapter which is focused on HLS at RTL.

The proposed RTL optimization flow assumes that the target architecture datapath is specified as a sequencing DFG, which is a directed acyclic DFG [30]. Each vertex of the DFG represents an operation, and each edge represents a data dependency. The DFG does not support hierarchical entities, and the conditional statements are handled using comparison operations. Also, each vertex has attributes that specify the operation type. The delay of a control step is dependent on the delays of the functional unit, the multiplexer, and register. The HLS flow also assumes that each node connected to the primary input is assigned two registers and one multiplexer while the inner nodes of the DFG have one register and one multiplexer. The SPICEless HLS flow generates gate-oxide leakage-optimal RTL description targeted for nano-CMOS technology. When the proposed behavioral scheduler is used along with the DT and propagation delay estimators, the system generates a circuit which dissipates minimal gate-oxide leakage power. The power and delay estimation phase uses analytical models introduced later in this chapter and calculates the values for different functional units [32, 45, 35]. It also calculates the total gate-oxide leakage power and critical path delay of the circuits when a scheduled DFG is given to it. In the above flow, it is assumed that a nano-CMOS integrated circuit is specified by the following: (1) a sequencing DFG, (2) a RTL library precharacterized for gate-oxide leakage and delay for nominal-oxide thickness devices and high-oxide thickness devices, (3) a set of resource constraints, and (4) a set of time constraints specified as a multiple of the critical path delay for the nominal-oxide thickness case.

9.5.2 Objective function for RTL optimization

The target architecture model for a digital integrated circuit is assumed to be as shown in Figure 9.7 [37, 28]. The digital integrated circuit datapath is assumed to be specified as a sequencing DFG. Each vertex of the DFG represents an operation, and each edge represents a dependency. Also, each vertex has attributes that specify the operation type. Each functional unit feeds one register and has a multiplexor also. The register

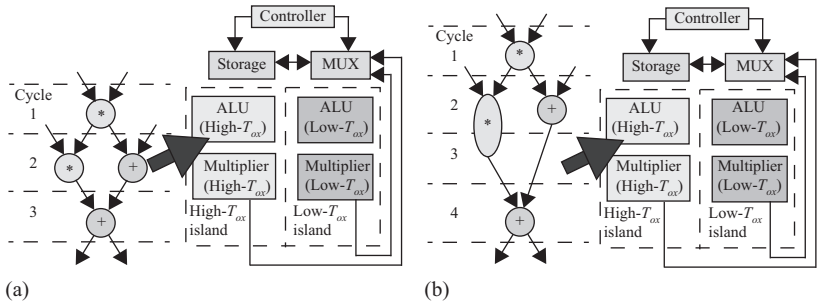


Figure 9.7 Scheduling, allocation, and binding during HLS for gate-oxide leakage-optimal target architecture [28]. (a) Single cycle architecture and (b) multicycle architecture

and the multiplexor belong to the same island (high-T_{ox} or low-T_{ox}) as that of the RTL units. A controller decides which functional units are active in each control step, and the inactive ones are disabled using the multiplexors. The delay of a control step is dependent on the delays of the functional unit, the multiplexer, and register. In one scenario shown in Figure 9.7(a), gate-oxide leakage reduction can be achieved by executing the addition in the second cycle in a high-T_{ox} ALU. In another scenario Figure 9.7(b), gate leakage reduction can be achieved by executing the multiplication in the second cycle in a high-T_{ox} multiplier.

The gate-oxide leakage optimization problem during architectural synthesis can be formalized as follows [28, 32, 35, 37]: *Given an unscheduled DFG (UDFG) $DFG_u(V, E)$, it is required to find the scheduled DFG (SDFG) $DFG_s(V, E)$ with appropriate resource binding such that the total gate leakage power dissipation is minimized and resource constraints (which is an indicator of silicon cost) and latency/timing/delay constraints (which is an indicator of integrated circuit performance) are satisfied.* The above can be stated as an optimization problem as follows. Let V be the set of all vertices, and V_{cp} be the set of vertices in the critical path from the source of the DFG to the sink vertex. The silicon cost (resource constrained) and performance (latency constrained) driven gate leakage minimization problem can thus be formulated as follows [28, 32, 35, 37]:

$$\text{Minimize} : \sum_{v_i \in V} P_{ox}(v_i), \tag{9.4}$$

where $P_{ox}(v_i)$ is the gate-oxide leakage dissipated per sample node v_i of the DFG, such that the specified resource and latency constraints are satisfied. The resource and delay constraints of the DFG can be represented respectively as follows:

$$\text{Allocated } (FU_i(k, T_{ox})) \leq \text{ Available } (FU_i(k, T_{ox})), \tag{9.5}$$

$$\sum_{v_i \in V_{cp}} T_{pd,i}(v_i) \leq DTF \times T_{pd,DFG}. \tag{9.6}$$

The constraints in (9.6) ensure that the summation of all delays $T_{pd,i}(v_i)$ is less than the specified time constraint, which is expressed as a multiple of the critical delay of the nominal case. The factor *DTF* is the time or performance trade-off factor. The resource allocation is summarized in (9.5), where the total allocation of the *i*th resources (functional units) of type k and made up of transistors of oxide thickness T_{ox} denoted as $(FU_i\,(k, T_{ox}))$ should be less than the total number of corresponding resources available.

The combined reduction of gate-oxide leakage power dissipation and execution time translates to reduction of the gate-oxide leakage and delay product (*LDP*). Thus, the objective of the scheduler is to minimize the *LDP* while assigning a schedule for the DFG. This implicitly facilitates minimization of tunneling current along with delay while considering resource constraints. Let us assume N_c- number of control steps and $n_{FU\,c}-$ number of resources active in any control step c. Then, the tunneling current-delay-product can be calculated as follows [28, 32, 35, 37]:

$$LDP = \sum_{c=1}^{N_c} \sum_{r=1}^{n_{FU\,c}} P_{ox\,FU}(c, r) \times T_{pd\,FU}(c, r). \tag{9.7}$$

In the above expression, $P_{ox\,FU}(c, r)$ is the tunneling current of the rth functional unit active in the control step c, and $T_{pd\,FU}(c, r)$ is its propagation delay. The scheduler generates various outputs, such as scheduled DFG with appropriate functional unit assignment to a datapath operation and estimates of current and delay. We assume that different functional units are characterized for tunneling current and propagation delay for various oxide thicknesses and are available in the component library. All the transistors inside the same resource have the same oxide thickness, and transistor gate-oxide thickness may differ for various functional units. However, to take process variation into account, we assume that a given gate-oxide thickness T_{oxp} can take any value in the range $\left(T_{oxp} - \Delta T_{oxp},\ T_{oxp} + \Delta T_{oxp}\right)$. It is assumed that such process variation is Gaussian [52]. It may be noted that we maintain constant $\left(\frac{L}{T_{oxp}}\right)$ and scale L along with T_{oxp}. Furthermore a constant $\left(\frac{W}{L}\right)$ ratio is maintained, as needed to maintain proper aspect ratio and reduce short-channel effects [47]. Thus, all three process parameters T_{oxp}, L, and W have Gaussian process variation.

9.5.3 A specific heuristic algorithm for RTL optimization

The key idea of the leakage optimization algorithm is presented in algorithm (Figure 9.8) [32, 35]. The algorithm takes in the datapath, specified as a sequencing DFG, which is a directed acyclic DFG, as an input. While each vertex of the DFG represents an operation, each edge represents a dependency. The DFG does not support hierarchical entities, and the conditional statements are handled using comparison operations. Each vertex has attributes to specify the operation type. From this input DFG along with the resource constraints, the algorithm determines the resource constrained ASAP (as soon as possible) and ALAP (as late as possible) schedules. In the next step, it identifies the critical path V_c and the off-critical paths V_{oc}. To begin

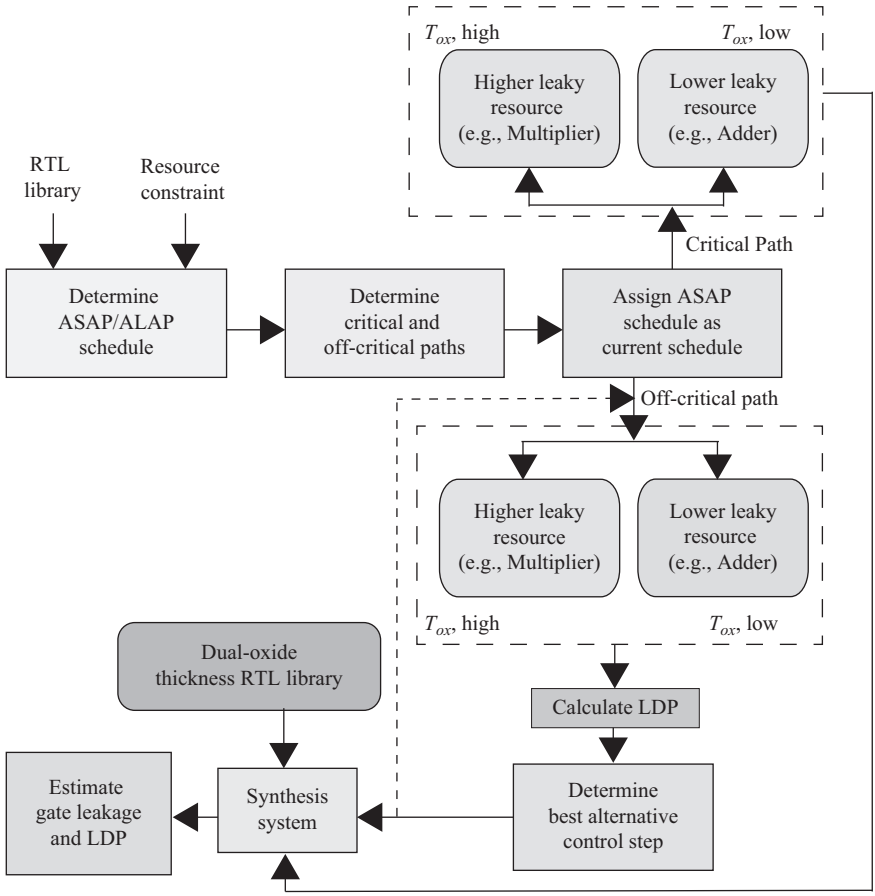


Figure 9.8 The key idea of the algorithm for gate-oxide leakage optimization during HLS

with, we consider the ASAP schedule as the default schedule. At this point, for each critical vertex V_c, we assign the largest gate-oxide thickness multiplier unit and smallest gate-oxide thickness adder–subtractor unit. The inputs to the behavioral scheduler are an unscheduled DFG, the resource constraints that include a number of different resources made of transistors of different oxide thickness. The scheduler time stamps the operations such that more low oxide thickness resources are active in the critical path and more high oxide thickness resources are active in the off-critical path of the datapath circuit. The scheduler attempts to assign higher intrinsic leakage functional units (such as multiplier and divider) with higher oxide thickness. This is in accordance with our conclusions from the analytical model where it is observed that multiplier and divider units dissipate much more tunneling current compared to adder and subtractor units. At the same time, it is observed that adder and subtractor units have lower delay compared to the multiplier and divider.

Thus, the scheduler attempts to operate the higher intrinsic leakage units of the highest thickness to reduce the tunneling current and at the same time lower intrinsic leakage units of lowest thickness to compensate the delay increase as much as possible. The same assignment is carried out in the case of all potential off-critical paths, and the *LDP* is calculated at each step and for each assignment for the DFG using (9.7). Once the minimum *LDP* is obtained, a particular vertex is time stamped and the T_{ox} assignment is accepted. The scheduler algorithm heuristic is presented in Algorithm 9.1 [32, 35].

In Algorithm 9.1, the first step is to determine the ASAP and ALAP schedules. While the ASAP schedule is unconstrained, the ALAP schedule uses the number of control steps found in the ASAP schedule as the latency constraint. The above obtained ASAP and ALAP schedules are modified using the resource constraints to determine the resource constrained ASAP and ALAP schedules. The resource constrained ASAP

Algorithm 9.1 Heuristic for dual-T_{ox} based leakage optimization during HLS [30, 32, 35]

1: **Input:** Unscheduled Data Flow Graph (UDFG), Resource Constraints, Timing Constraints in the form of delay trade-off factor (DTF), Analytic Functions for P_{oxFU} and $T_{pd_{FU}}$.
2: **Output:** Scheduled Data Flow Graph (SDFG), Gate-Oxide Leakage Power and Delay Estimates, Number of Clock Cycles.
3: Determine the total number of RTL components of all available gate-oxide thicknesses from the DFG: $G(V, E)$.
4: Determine the resource constrained as soon as possible schedule S_{ASAP}.
5: Determine the resource constrained as late as possible sch edule S_{ALAP}.
6: Fix the total number of clock cycles or time stamps as the maximum of S_{ASAP} and S_{ALAP} steps.
7: Assume the above S_{ASAP} schedule as the current schedule S_i.
8: Find the vertices in critical path V_c and off-critical path V_{oc} (where, both V_c and $V_{oc} \in V$) of the DFG.
9: **for** (Each of the critical path vertices, i.e., $v \in V_c$) **do**
10: Assign highest thickness T_{oxH} to operations needing high-leaky resources such as multipliers and dividers.
11: Lowest thickness T_{oxL} to operations needing low-leaky resources such as adders.
12: **end for**
13: **while** (All $v \in V_{oc}$ of the current schedule S_i are not considered for time stamping)**do**
14: **if** (A vertex v is needs a high-leaky RTL component) **then**
15: Assign the highest available thickness high-T_{ox}.
16: **else**
17: Assign the lowest available thickness low-T_{ox}.
18: **end if**

Algorithm 9.1 Continued

19: Generate Gaussian random numbers in the range of $(T_{ox} - \Delta T_{ox}, T_{ox} + \Delta T_{ox})$ to take the process variations into consideration during the estimation of the characteristics of the target integrated circuit architecture.
20: Calculate the leakage delay product of the current schedule LDP_{S_i} for one value from the range $(T_{ox} - \Delta T_{ox}, T_{ox} + \Delta T_{ox})$ using the analytical functions representing the characteristics of the RTL components.
21: **for** (Each off-critical vertex V_{oc} (i.e., $v \in V_{oc}$) of the current schedule S_i) **do**
22: **for** (Every allowable control step c in the mobility range of v) **do**
23: Assign next higher thickness if vertex needs high leaky resource and next lower thickness if vertex needs low leaky resource.
24: Generate Gaussian random numbers in the range of $(T_{ox} - \Delta T_{ox}$, $T_{ox} + \Delta T_{ox})$.
25: Find *LDP* of the DFG for each case for a values from $(T_{ox} - \Delta T_{ox}$, $T_{ox} + \Delta T_{ox})$.
26: **end for**
27: **end for**
28: Fix the clock cycle of the vertex with the current T_{ox} assignment for which *LDP* is minimum.
29: Remove the above time stamped vertex v from V_{oc} and mark it as scheduled.
30: **end while**
31: Determine all vertices scheduled in each clock cycle.
32: **for** (Particular type of operations in a clock cycle) **do**
33: **if** (A critical vertex has higher T_{ox} than a off-critical vertex) **then**
34: Swap the specific T_{ox} RTL components.
35: **end if**
36: **end for**
37: Calculate leakage power dissipation and critical path delay for the scheduled DFG.
38: **return** Scheduled DFG, estimates of leakage power dissipation and critical path delay.

and ALAP schedules restrict the mobility of vertices to a great extent and reduce the solution search space for the heuristic. In the next step, the algorithm identifies the critical path V_c and the off-critical paths V_{oc}. Vertices with the same ASAP and ALAP time stamps (i.e., zero mobility) are the critical vertices which are needed to be given more priority over off-critical vertices. For each critical vertex V_c, the algorithm assigns the highest gate-oxide thickness to the operation needing higher leakage units, while we assign the lowest available gate-oxide thickness to the operations needing low leakage units. In this way the performance loss due to high leakage, high-oxide thickness (high-T_{ox}) resources would be compensated by the low leakage, low-oxide thickness (low-T_{ox}) resources. The algorithm attempts to find suitable time stamp,

gate-oxide thickness for the off-critical vertices using an exhaustive search. The off-critical vertices are attempted to be placed in each of the control steps within their allowable mobility range. During each placement, gate-oxide thickness assignment is done and *LDP* value is estimated. The predecessor and successor time clock cycles are adjusted accordingly to maintain the data dependency. A particular vertex is time stamped for a clock cycle with a T_{ox} assignment for which *LDP* is minimum. Gaussian distributed random numbers are generated to take into account the effect of process variation on T_{ox}; the values are generated in the range $(T_{ox} - \Delta T_{ox},\ T_{ox} + \Delta T_{ox})$. The algorithm picks any one value in that range to replace T_{ox} under consideration. The algorithm in the final step scans through every clock cycle and finds all the scheduled vertices in each. For a particular type of operation if the critical vertex has higher T_{ox} than an off-critical vertex then the values of T_{ox} are swapped between them. This step further compensates the performance degradation due to the use of high leakage resources with higher T_{ox}. The above described algorithm can be easily used to handle various types of datapath operations, such as multicycling, chaining, and pipelining. For example, to take the multicycling operation into account, the algorithm can assume the delay of the fastest unit as clock width and time stamp vertices needing slower unit to more than one clock steps.

9.6 SPICEless characterization of the RTL component library

This section discusses the detailed process of SPICEless characterization of the RTL component library. The characterization data can be either represented in tabular form or analytical models. Analytical models are preferred in this chapter for easy calculation of data accounting for process variations. The analytical models for gate-oxide leakage and propagation delay calculation are presented for architecture-level functional units. In the absence of foundry rules, these models are useful in characterizing the RTL components for design space exploration. Such models completely bridge the architectural-level abstraction with the transistor level and help in quicker decision making at architecture level before laying out the design in silicon. A top-down design synthesis with a three-level hierarchy is used to form the models. At the top level of hierarchy, the RTL components such as adders, subtractors, multipliers, etc. are present. They in turn make use of logic-level components which are derived from a set of equations available for various transistor characteristics. The models are developed from first principles using standard equations considering various physical effects, such as polysilicon depletion, channel length modulation, mobility degradation, velocity saturation, and body bias. Finally, the gate-oxide leakage and propagation delay of each RTL unit is calculated in terms of gate-oxide thickness for different technologies in order to facilitate the behavioral synthesis process. The steps of RTL component library characterization are presented in Algorithm 9.2. The generation of NAND netlist and logic-level optimization can be based on earlier logic-level optimization research [47]. In the current chapter, first principle physics based models are used for logic gates. However, a SPICEless alternative is the use of MATLAB®/Simulink® simulations [19].

Algorithm 9.2 Steps for SPICEless characterization of RTL component library

1: **Input:** Components of RTL library.
2: **Output:** Analytical models of target characteristics of RTL component library.
3: Obtain logic level netlist of the RTL components.
4: Perform logic-level optimization of the netlist.
5: Obtain NAND based netlist of the RTL components as NAND has lowest leakage among the logic gates and is an universal gate.
6: Characterize NAND gates using first-principle physics models.
7: Characterize RTL component library using NAND netlist and NAND characterization information.
8: Perform known function fitting to RTL characterization data to obtain analytical models.
9: **return:** Analytical models of target characteristics of RTL component library.

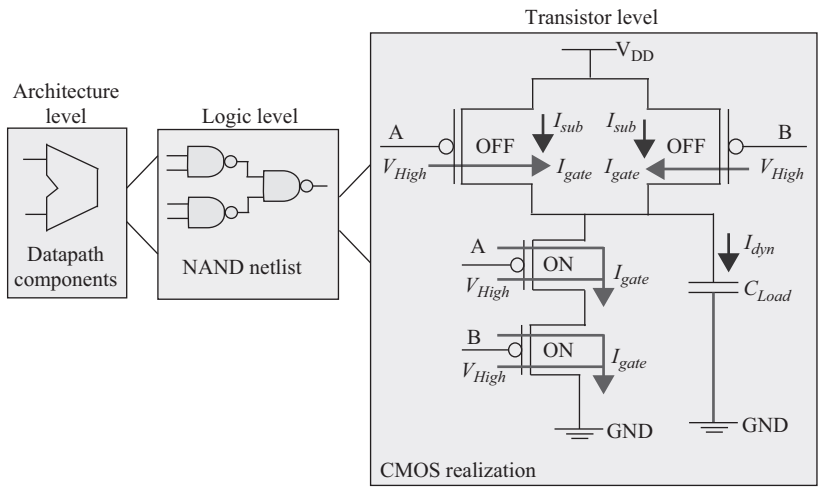


Figure 9.9 Three-level hierarchical representation for component modeling [31, 35]

In the hierarchical modeling of the RTL components, it is assumed that datapath units are constructed using universal logic gates, such as the NAND. The three-level hierarchical representation for RTL component modeling is shown in Figure 9.9 [31, 35]. Prior research suggests that NAND gates have minimal gate-oxide leakage compared to other logic gates [47]. Let us assume that there are total n_{total} NAND gates in the network of NAND gates constituting a n-bit functional unit out of which n_{cp} number of NAND gates are in the critical path. In this model, the effect of interconnect is not considered, and focus is on the gate-oxide leakage power dissipation and propagation delay of the functional units only. It may be noted that this assumption

does not affect the gate-oxide leakage values as gate-oxide tunneling occurs only in the active devices not in the interconnects. It is assumed that all transistors used in a functional unit have gate oxide of equal thickness, but the thickness of different functional units may differ.

9.6.1 Gate-oxide leakage modeling

The first-principle physics models for calculation of gate-oxide leakage of logic are now presented. The gate-oxide tunneling mechanism in a CMOS can be either FN tunneling or DT; both differ in the form of potential barrier [55]. We consider the tunneling to be direct with trapezoidal potential barrier. The tunneling probability of an electron is affected by barrier height, structure, and thickness of the barrier material and is predominant for thinner gate oxide, which is the case for sub-65 nm technology.

The gate-oxide leakage of an n-bit architecture-level functional unit can be estimated as follows [32, 35]:

$$P_{ox}\mathrm{FU} = \sum_{j=1}^{n_{total}} Pr_j \sum_{\mathrm{MOS}_i \in \mathrm{NAND}_j} Pr_i \, P_{oxi}. \tag{9.8}$$

In the above expression, Pr_j is the probability that the input of the NAND gate is at logic "0", which can be obtained by logic-level estimations. The contributions of the NMOS and PMOS tunneling depend on the probability of the input signal being at logic "1" and "0", respectively. The average tunneling current for a logic gate is calculated as [32, 35]:

$$P_{ox}\mathrm{NAND} = \sum_{\mathrm{MOS}_i \in \mathrm{NAND}} Pr_i \, P_{oxi}, \tag{9.9}$$

where Pr_i is the probability that inputs of the MOS that are connected in parallel (i.e., PMOS) are at logic "0". The situation for a logic gate is more complex than a single device since the internal state of the gate and its overall response depend on the values of a number of inputs. The current paths for a two-input NAND gate is illustrated in Figure 9.10. The figures also include the internal states of the transistors as well as the gate-oxide tunneling current paths for all possible combinations of inputs to the logic gate [34, 44]. V_{Th} voltage drops have not been taken into account.

The models for P_{ox} for a NAND gate which are used to calculate the tunneling current of FUs are now presented. The gate-oxide leakage is a result of DT with trapezoidal potential barrier. The tunneling probability of an electron is affected by barrier height, structure and thickness and is expressed by the following [11, 14, 32, 35, 55]:

$$I_{ox,}\mathrm{MOS} = \left(\frac{WL\, q^3 V_{ox}^2}{16\pi^2 \hbar \phi_B T_{ox}^2}\right) \exp\left[-\frac{4\sqrt{2m_{eff}}\, {\phi_B}^{1.5} T_{ox}}{3\hbar q V_{ox}} \left\{1 - \left(1 - \frac{V_{ox}}{\phi_B}\right)^{1.5}\right\}\right]. \tag{9.10}$$

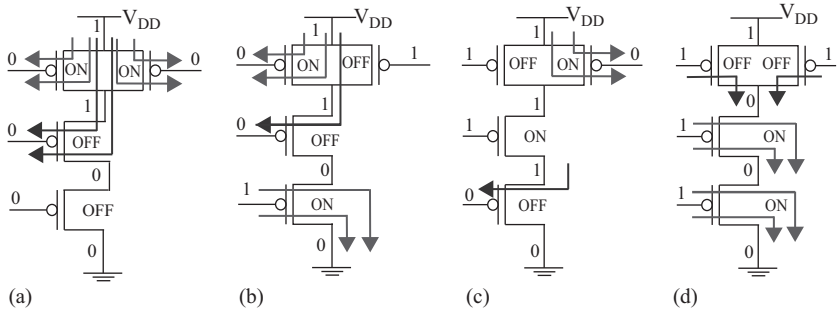


Figure 9.10 Gate-oxide leakage current paths in various states of a two-input NAND [34, 44]. (a) For Input 00, (b) for Input 01, (c) for Input 10, and (d) for Input 11

The voltage across the MOS gate dielectric V_{ox} is expressed as follows [32, 35, 55, 69]:

$$V_{ox} = \left(V_{gs} - V_{fb} - \psi_S - V_{poly}\right). \tag{9.11}$$

In the above expression, the voltage across the polysilicon depletion region V_{poly} is expressed as follows [32, 35, 55]:

$$V_{poly} = \left(\frac{\varepsilon^2{}_{ox}\, V_{ox}^2}{2q\ \varepsilon_{Si}\ N_{poly} T_{ox}^2}\right) \tag{9.12}$$

From these two equations, we obtain a quadratic equation in terms of the variable V_{ox}. By solving this quadratic equation, the following expression is obtained for V_{ox} [32, 35]:

$$V_{ox} = \left(\frac{\sqrt{1 - 2(V_{fb} + \psi_S - V_{gs})\left(\frac{\varepsilon^2{}_{ox}}{q\ \epsilon_{Si}\ N_{poly} T_{ox}^2}\right)} - 1}{\left(\frac{\varepsilon^2{}_{ox}}{q\ \varepsilon_{Si}\ N_{poly} T_{ox}^2}\right)}\right). \tag{9.13}$$

The flat-band voltage V_{fb} can be derived from MOS capacitance–voltage (C–V) characteristics or using the following expression [32, 35]:

$$V_{fb} = \left(\frac{qN_{channel} T_{ox}^2}{2\varepsilon_{Si}}\right). \tag{9.14}$$

It may be noted that the effective values of W, L, may be different from the nominal values due to depletion and need to be taken into consideration [8, 73]. The effective gate-oxide thickness T_{ox} is a quadratic function of the physical oxide thickness, T_{oxp}

[8, 73]. After solving the quadratic equation and taking polysilicon depletion into consideration, the following expression for the effective thickness is obtained [32, 35]:

$$T_{ox} = 0.5T_{oxp}\left(1 + \sqrt{1 + 4\frac{\varepsilon_{ox}}{\varepsilon_{Si}}\frac{X_{poly}}{T_{oxp}}}\right). \tag{9.15}$$

The gate polysilicon depletion depth is calculated using the following expression [8, 32, 35]:

$$X_{poly} = \frac{\varepsilon_{ox}}{\varepsilon_{Si}}T_{oxp}\left(\sqrt{1 + \frac{2\varepsilon^2{}_{ox}(V_{gs} - V_{fb} - \psi_S)}{q\varepsilon_{Si}N_{poly}T^2{}_{oxp}}} - 1\right). \tag{9.16}$$

The Fermi-level ϕ_F is calculated using the following expression [32, 35]:

$$\phi_F = \left[\frac{2kT}{q}\ \ln\left(\frac{N_{channel}}{n_i}\right)\right]. \tag{9.17}$$

In the above expression, it is assumed that strong inversion is taking place in the MOSFET and the surface potential ψ_S is $2\phi_F$ [14, 67, 68].

9.6.2 Propagation delay modeling

The models that will be used for propagation delay calculation of functional units of a datapath are now discussed. The critical path delay of an n-bit functional unit using the above NAND gates as building blocks can be calculated as follows [32, 35]:

$$T_{pd}\mathrm{FU} = \sum_{i=1}^{n_{cp}} 0.5\left(n_{fan-in}T_{pd}\mathrm{NMOS} + T_{pd}\mathrm{PMOS}\right). \tag{9.18}$$

The effective fan-in factor, $n_{fan\text{-}in}$, is calculated for short-channel devices with velocity saturation and strong inversion using the following expression [3, 5, 32, 35]:

$$n_{fan-in} = 1 + \left[\frac{\left(2 - \sqrt{2}\right)(n_{series} - 1)\,V_{dsSat}}{V_{dd} + V_{Th} - 0.5V_{dsSat}}\right]\left(1 + \frac{T_{ox}}{\varepsilon_{ox}}\sqrt{\frac{qN_{channel}\varepsilon_{Si}}{2\psi_S}}\right), \tag{9.19}$$

where n_{series} is the number of series connected MOS devices.

The α-power law and physical-α-power model are used to compute the propagation delay (T_{pd}) of a MOS as follows [7, 8, 32, 35, 56]:

$$T_{pd} = \frac{0.5C_LV_{dd}}{I_{DSat0}} + T_T\left[\frac{0.5 - \left(\frac{V_{dd} - V_{Th}}{V_{dd}}\right)}{\alpha + 1}\right]. \tag{9.20}$$

In the above expression, I_{DSat0} is the saturation drain current of the MOS for $V_{gs} = V_{dd}$. The saturation drain current is calculated by the following expression [8, 32, 35]:

$$I_{DSat} = \frac{W}{L}\left(\frac{V_{gs} - V_{Th}}{V_{dd} - V_{Th}}\right)^{\alpha}\left[\frac{\mu_0 C_{ox} V_{dsSat0}\left(V_{dd} - V_{Th} - 0.5\eta V_{dsSat0}\right)}{\left(1 + \theta\left(V_{gs} - V_{Th}\right)\right)\left(1 + \frac{\mu_0 V_{dsSat}}{v_{sat} L\left(1+\theta\left(V_{gs} - V_{Th}\right)\right)}\right)}\right]. \quad (9.21)$$

The zero bias mobility can be calculated using the following expression [15, 32, 35, 67]:

$$\mu_0 = \frac{\mu_{sub}}{\left[1 + \left(\frac{Q_B \mu_{sub}}{\varepsilon_{ox} v_{norm}}\right)\right]}. \quad (9.22)$$

In the above expression, the depletion charge density Q_B is calculated using the following expression [15, 32, 35, 67]:

$$Q_B = \sqrt{2q\varepsilon_{Si} N_{sub} \psi_S}. \quad (9.23)$$

The transition time model is expressed as follows [8, 32, 35]:

$$T_T = \frac{C_L V_{dd}}{I_{DSat0}}\left[\frac{0.9}{0.8} + \frac{V_{dsSat0}}{0.8 V_{dd}}\left(\frac{V_{dd} - V_{Th} - 0.5\eta V_{dsSat0}}{V_{dd} - V_{Th}}\right.\right.$$
$$\left.\left. \times ln\left(\frac{10 V_{dsSat0}\left(V_{dd} - V_{Th}\right)}{V_{dd}\left(V_{dd} - V_{Th} - 0.5\eta V_{dsSat0}\right)}\right) - 1\right)\right]. \quad (9.24)$$

The constant α, modeling carrier saturation velocity is calculated as follows [7, 8, 32, 35]:

$$\alpha = \frac{1}{\ln(2)} \ln\left[\frac{2V_{dsSat0}\left(V_{dd} - V_{Th} - 0.5\eta V_{dsSat0}\right)}{V_{dsSata}\left(V_{dd} - V_{Th} - \eta V_{dsSata}\right)}\right]. \quad (9.25)$$

In the above expression, V_{dsSat0} and V_{dsSata} are the saturation drain voltages for $V_{gs} = V_{dd}$ and $V_{gs} = \left(\frac{V_{dd}+V_{Th}}{2}\right)$, respectively. The saturation drain voltage V_{dsSat} is given by the following expression [8, 7, 32, 35]:

$$V_{dsSat} = \frac{v_{sat} L}{\mu_0}\left(1 + \theta\left(V_{gs} - V_{Th}\right)\right)\left[\sqrt{1 + \frac{2\mu_0\left(V_{gs} - V_{Th}\right)}{v_{sat} L \eta\left(1 + \theta\left(V_{gs} - V_{Th}\right)\right)}} - 1\right]. \quad (9.26)$$

The effective threshold voltage in all of the above equations is calculated using (9.27) [32, 35, 55]:

$$V_{Th} = V_{fb} + \frac{2kT}{q}\ln\left(\frac{N_{sub}}{n_i}\right) + \frac{1}{C_{ox}}\sqrt{2q\varepsilon_{Si} N_{sub}\left(\frac{2kT}{q}\ln\left(\frac{N_{sub}}{n_i}\right) + V_{bs}\right)}, \quad (9.27)$$

where the effective oxide thickness for the C_{ox} calculation is performed assuming strong inversion. The mobility degradation factor θ is calculated using the following expression [7, 8, 32, 35]:

$$\theta = \left(\frac{\mu_0}{2T_{ox} v_{norm}} \right), \tag{9.28}$$

where T_{ox} is calculated using (9.15). The subthreshold slope factor η is calculated using the following expression [7, 8, 32, 35]:

$$\eta = \left[1 + \sqrt{\frac{q\varepsilon_{Si} N_{channel} T_{ox}^2}{2\varepsilon_{ox}^2 (\psi_S - V_{bs})}} \right]. \tag{9.29}$$

9.6.3 Analytical modeling of RTL components

In order that the above models become useful for dual-T_{ox} RTL optimization for gate-oxide leakage reduction, the characteristics need to be expressed in terms of functions of T_{ox}. It is assumed that the functional units, such as adder, subtractor, multiplier, divider, and comparator are of 16-bit size. The structural information for them is obtained from Reference 71. The units are presented in the form of NAND gates and are characterized using the models presented in the previous subsections. The parameters used for the calculation are shown in Table 9.2. These values are obtained based on various published data in the existing literature [9, 67, 68]. It is assumed

Table 9.2 Specific values of parameters used in modeling

Device parameters	Specific values used in modeling
Bulk mobility $\mu_{subNMOS}$	750 $\frac{cm^2}{V-s}$
Bulk mobility $\mu_{subPMOS}$	250 $\frac{cm^2}{V-s}$
Permittivity of gate dielectric ε_{ox}	3.9 $\varepsilon_o \frac{F}{m}$
Barrier height for the gate dielectric ϕ_B	3.15 eV
Constant for mass calculation k_{mNMOS}	0.19
Constant for mass calculation k_{mPMOS}	0.55
Intrinsic concentration n_i	9.5×10^9 per cm^3
Proportionality constant v_{norm}	2.2×10^9 $\frac{cm}{s}$
Electron saturation velocity v_{sat}	6.4×10^6 $\frac{cm}{s}$
Channel doping concentrations $N_{channel}$	1.7×10^{17} per cm^3
Polysilicon gate doping concentrations N_{poly}	5.0×10^{19} per cm^3
Substrate doping concentrations N_{sub}	6.0×10^{16} per cm^3
Temperature T	300 K
Supply voltage V_{dd}	1.0 V
Gate voltage V_{gs}	1.0 V
Threshold voltage V_{ThNMOS}	0.22 V
Threshold voltage V_{ThPMOS}	−0.22 V
Body-to-source voltage V_{bs}	0 V
Flat-band voltage V_{fb}	−0.55 V

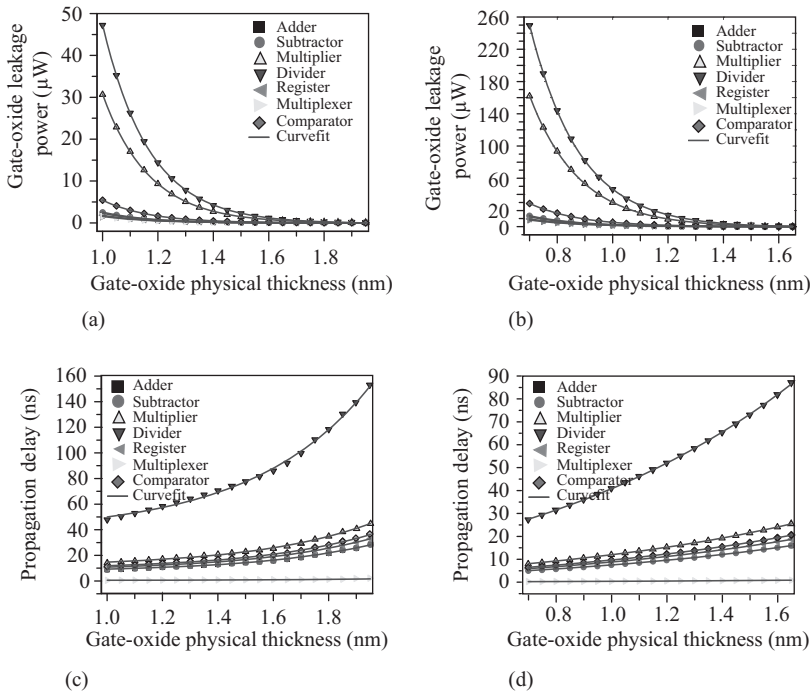


Figure 9.11 Gate-oxide leakage power current and propagation delay versus oxide physical thickness for 65 nm and 45 nm technologies. (a) Gate leakage for 65 nm CMOS, (b) gate leakage for 45 nm CMOS, (c) propagation delay for 65 nm CMOS, and (d) propagation delay for 45 nm CMOS

that the probability of logic "1" and logic "0" is the same. For a given length L, the width of the transistors is chosen as $W_{NMOS} = 4L$, $W_{PMOS} = 8L$ to ensure smooth current flow between NMOS and PMOS. While changing the oxide thickness the channel length of the transistor is changed proportionately to avoid impact on its functionality [65]. In other words, as the T_{oxp} value is increased, the length is changed as $L_{new} = \left(\frac{L}{T_{oxp}}\right) T_{oxp_{new}}$. This constant aspect ratio of $\left(\frac{L}{T_{ox}}\right)$ ensures constant per width gate capacitance of the transistor, as per fabrication requirements [47, 71].

The various analytical models are pictorially presented in Figure 9.11. The changes of gate-oxide leakage dissipation of the functional units as the physical oxide thickness (T_{oxp}) changes are shown in for Figure 9.11(a) and 9.11(b) for 65 nm and 45 nm technology, respectively. Similarly, the changes of propagation delay of the functional units as the physical oxide thickness (T_{oxp}) changes are shown in for Figure 9.11(c) and 9.11(d) for 65 nm and 45 nm technology, respectively. For 65 nm technology, the gate-oxide physical thickness is varied in the range of $T_{oxp} = 1.0\,\text{nm} \rightarrow 1.95\,\text{nm}$ in an increment of 0.05 nm. In such a range of values, the following values are used $L = 65\,\text{nm} \rightarrow 126.75\,\text{nm}$, $W_{NMOS} = 260\,\text{nm} \rightarrow 507\,\text{nm}$,

and $W_{PMOS} = 520\,\text{nm} \rightarrow 1014\,\text{nm}$. For 45 nm technology, the following range of values are used: $T_{oxp} = 0.7\,\text{nm} \rightarrow 1.65\,\text{nm}$, $L = 45\,\text{nm} \rightarrow 106.1\,\text{nm}$, $W_{NMOS} = 180\,\text{nm} \rightarrow 424.3\,\text{nm}$, and $W_{PMOS} = 360\,\text{nm} \rightarrow 848.66\,\text{nm}$. The plots shown in Figure 9.11(c) and 9.11(d) confirm that there is drastic reduction in the gate-oxide leakage as the oxide thickness increases. At the same time, there is increase in the propagation delay for the functional unit as evident from Figure 9.11(c) and 9.11(d). It is also observed that there is increase in the gate-oxide leakage as technology scales from 65 nm to 45 nm, which is consistent with the ITRS prediction trend [1]. The gate-oxide leakage and propagation delay of various RTL components are expressed in terms of gate-oxide thickness in order to use them in the proposed low power HLS flow. The analytical functions are presented in Table 9.3 for various units for different technologies [32, 35].

9.7 Experimental results for the specific RTL optimization

The algorithm was implemented for experiments in an in-house behavioral synthesis framework [30, 38]. It is tested with several behavioral-level benchmark circuits for several constraints. In this section, the experimental results are presented for selected benchmarks and constraints. A selected set of HLS benchmarks used are as follows [30, 32, 35, 42, 45]:

- Auto-regressive filter (ARF) (total 28 nodes, 16*, 12+, 40 edges),
- Band-pass filter (BPF) (total 29 nodes, 10*, 10+, 9−, 40 edges),
- Discrete-cosine transformation (DCT) filter (total 42 nodes, 13*, 29+, 68 edges),
- Elliptic-wave filter (EWF) (total 34 nodes, 8*, 26+, 53 edges),
- Finite impulse response (FIR) filter (total 23 nodes, 8*, 15+, 32 edges), and
- HAL differential equation solver (HAL) (total 11 nodes, 6*, 2+, 2−, 1<, 16 edges)

The above are DSP benchmarks. These benchmarks are very computationally intensive and a perfect choice for custom hardware design instead of software for low-cost and high-performance system design purposes. However, any other benchmarks can be used in the proposed RTL optimization flow through HLS.

In the first phase of the experiments, resources of two different gate-oxide thicknesses are used for dual-T_{ox}. For both 65 nm and 45 nm technology, two different oxide thicknesses are selected in which the higher thickness is 35% more than the lower thickness. A selected set of resource constraints are given in Table 9.4 for resources of two different gate-oxide thicknesses. These represent the functional units of different thicknesses available to the behavioral scheduler. The sets of resource constraints were chosen so as to cover functional units consisting of different oxide thicknesses. These are representatives of various forms of the corresponding RTL representation. The number of dividers or comparators is not shown in the table as there was only one benchmark (HAL) that needed a comparator, and there were no benchmarks in the above listed ones that needed a divider.

Table 9.3 Analytical models in terms of T_{oxp} to be used for dual-T_{ox} based RTL optimization

	Gate-oxide leakage in μW, $P_{ox\text{FU}} = f(T_{oxp})$.						**Propagation delay in ns, $T_{pd_{\text{FU}}} = g(T_{oxp})$.**					
	$f(T_{oxp}) = \alpha e\left(-\frac{T_{oxp}}{\gamma}\right) + \beta$						$g(T_{oxp}) = \alpha e\left(\frac{T_{oxp}}{\gamma}\right) + \beta$					
	65 nm technology			**45 nm technology**			**65 nm technology**			**45 nm technology**		
	γ	α	β	γ	α	β	γ	α	β	γ	α	β
Adder	0.17	8.64×10^{2}	-7.54×10^{-3}	1.80	5.93×10^{2}	-5.39×10^{-2}	0.42	0.21	6.98	1.05	3.81	−2.33
Subtractor	0.17	9.66×10^{2}	-8.43×10^{-3}	1.80	6.63×10^{2}	-6.02×10^{-2}	0.42	0.21	6.98	1.05	3.81	−2.33
Multiplier	0.17	1.15×10^{4}	-1.00×10^{-1}	1.80	7.92×10^{3}	-7.19×10^{-1}	0.42	0.34	11.10	1.05	6.07	−3.71
Divider	0.17	1.78×10^{4}	-1.55×10^{-1}	1.80	1.22×10^{4}	$-1.11 \times 10^{+0}$	0.42	1.16	37.8	1.05	20.60	−12.61
Comparator	0.17	2.05×10^{3}	-1.79×10^{-2}	1.80	1.41×10^{3}	-1.28×10^{-1}	0.42	0.28	8.96	1.05039	4.89	−2.99
Register	0.17	6.86×10^{2}	-5.99×10^{-3}	1.80	4.71×10^{2}	-4.28×10^{-2}	0.42	0.25	8.17	1.05	4.46	−2.73
Multiplexer	0.17	5.84×10^{2}	-5.09×10^{-3}	1.80	4.01×10^{2}	-3.64×10^{-2}	0.42	0.01	0.40	1.05	0.22	−0.13

Table 9.4 A selected list of resource constraints used to perform our experiments

Number of resources for various T_{oxp}												Resource constraint number
65 *nm* technology						45 *nm* technology						
Multiplier		Adder		Subtractor		Multiplier		Adder		Subtractor		
1.35 nm	1.0 nm	1.35 nm	1.0 nm	1.35 nm	1.0 nm	0.95 nm	0.7 nm	0.95 nm	0.7 nm	0.95 nm	0.7 nm	
1	1	2	0	2	0	1	1	2	0	2	0	1
2	1	1	1	1	1	2	1	1	1	1	1	2
2	0	0	2	0	2	2	0	0	2	0	2	3

Table 9.5 RTL optimization results of various digital benchmark circuits for 65 nm technology

Benchmark digital circuits	Resource constraints	P_{ox} in μW			T_{pd} in ns		
		ST	***DT***	**ΔP_{ox}**	***ST***	***DT***	**ΔT_{pd}**
ARF	1	521.2	251.9	51.7	142.1	190.0	33.6
	2	521.2	161.8	69.0	142.1	167.0	17.5
	3	521.2	89.8	82.8	142.1	174.2	22.5
		Average ΔP_{ox}		71.96	Average ΔT_{pd}		22.81
BPF	1	411.07	157.69	61.63	127.93	169.96	32.85
	2	411.07	123.71	69.90	127.93	159.96	25.04
	3	411.0	87.5	78.7	127.9	154.2	20.5
		Average ΔP_{ox}		73.3	Average ΔT_{pd}		25.9
DCT	1	472.0	84.2	39.4	213.2	269.9	26.6
	2	472.0	84.2	82.2	213.2	269.9	26.6
	3	472.0	121.5	74.2	213.2	261.3	22.5
		Average ΔP_{ox}		79.8	Average ΔT_{pd}		25.3
EWF	1	311.0	37.7	87.9	227.4	250.8	10.3
	2	311.0	70.9	77.2	227.4	239.9	5.5
	3	311.0	95.3	69.3	227.4	233.6	2.7
		Average ΔP_{ox}		77.9	Average ΔT_{pd}		6.0
FIR	1	283.3	115.2	59.3	156.3	183.2	17.2
	2	283.3	58.7	79.2	156.3	180.0	15.2
	3	283.3	67.6	76.1	156.3	159.9	2.3
		Average ΔP_{ox}		73.5	Average ΔT_{pd}		12.4
HAL	1	196.7	77.8	60.4	56.8	80.0	40.7
	2	196.7	59.7	69.6	56.8	67.1	18.0
	3	196.7	34.9	82.2	56.8	67.1	18.0
		Average ΔP_{ox}		73.9	Average ΔT_{pd}		20.6
Overall		Average ΔP_{ox}		75.1	Average ΔT_{pd}		18.1

The experimental results for various benchmark digital circuits are presented in Table 9.5 for 65 nm technology for dual-T_{ox} technique [32]. The quantities with ST subscript represent results for single thickness and DT subscript represent results for the multiple oxide thickness case. We assume the minimal oxide thickness case with T_{oxp} of 0.7 nm as the base ST case. The value of ΔT_{oxp} is assumed to be 10% of the original T_{oxp}. The percentage reduction in gate-oxide leakage is calculated as follows: $\Delta P_{ox} = \left(\frac{P_{oxST} - P_{oxDT}}{P_{oxST}}\right) * 100\%$. It is also evident that there is an increase in delay as the gate-oxide thickness increases. So, the percentage time penalty is calculated as follows: $\Delta T_{pd} = \left(\frac{T_{pdDT} - T_{pdST}}{T_{pdST}}\right) * 100\%$.

It is observed from the experimental results from Table 9.5 that the reduction in gate-oxide leakage is in the range of 51.6–87.9% with an overall average of 75.0%. The average reduction for each benchmark is very consistent and in the range of 71.9–79.9%. From the results table it can be seen that the reduction in gate-oxide leakage

Table 9.6 RTL optimization results of various digital benchmark circuits for 45 nm technology

Benchmark digital circuits	Resource constraints	P_{ox} in μW			T_{pd} in ns		
		ST	***DT***	**ΔP_{ox}**	***ST***	***DT***	**ΔT_{pd}**
ARF	1	1360.5	647.8	52.4	34.9	49.7	42.6
	2	1360.5	409.2	69.9	34.9	43.8	33.0
	3	1360.5	218.6	83.9	34.8	45.7	31.2
	Average ΔP_{ox}			73.0	Average ΔT_{pd}		35.0
BPF	1	1073.0	402.3	62.5	31.0	44.5	43.5
	2	1073.0	312.4	70.9	29.0	41.9	44.1
	3	1073.0	216.6	79.8	30.9	40.5	30.7
	Average ΔP_{ox}			74.3	Average ΔT_{pd}		40.6
DCT	1	1232.1	205.6	83.3	52.3	70.6	35.1
	2	1232.1	222.2	82.0	52.3	69.9	33.8
	3	1232.1	304.3	75.3	52.3	68.6	31.2
	Average ΔP_{ox}			80.6	Average ΔT_{pd}		33.5
EWF	1	811.9	88.4	89.1	44.5	62.8	41.0
	2	811.9	176.4	78.3	44.5	58.1	30.5
	3	811.9	240.9	70.3	44.5	54.0	21.3
	Average ΔP_{ox}			79.0	Average ΔT_{pd}		30.8
FIR	1	739.5	294.6	60.1	29.0	41.7	44.1
	2	739.5	145.0	80.3	29.0	35.1	21.0
	3	739.5	168.5	77.2	29.0	34.5	18.7
	Average ΔP_{ox}			74.5	Average ΔT_{pd}		22.2
HAL	1	513.5	198.7	61.3	13.5	20.9	54.4
	2	513.5	150.7	70.6	11.6	17.6	51.7
	3	513.5	85.2	83.4	13.5	17.6	30.11
	Average ΔP_{ox}			74.7	Average ΔT_{pd}		41.4
Overall		Average ΔP_{ox}		76.02	Average ΔT_{pd}		34.58

is maximum for the DCT and EWF benchmarks and minimum for ARF benchmark. The critical path delay of the circuit is calculated as the sum of the delays of the vertices in the longest path of the DFG, which has been reported in the results table. The time penalty is found to be in the range of 6.0–25.8% with an average overall average of 18.8%. The experimental results for the 45 nm technology for the same set of benchmark circuits are more or less similar to that of 65 nm technology. The detailed experimental results are presented in Table 9.6 [35]. The reduction in the gate-oxide leakage is decreased by approximately 10–12%. However, the average time penalty for both 45 nm and 65 nm technologies is approximately equal. The average experimental results are presented in Figure 9.12 for a comparative perspective of 65 nm and 45 nm technologies for various benchmark circuits [32, 35].

The dual-T_{ox} technique uses RTL components made of two difference oxide thicknesses. However, it is possible to use more than two oxide thicknesses, at least in

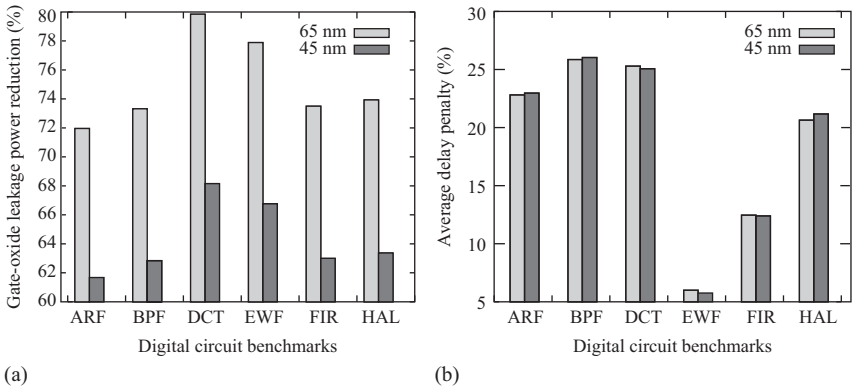


Figure 9.12 Average results for various HLS digital integrated circuit benchmarks. (a) For leakage reduction and (b) for delay penalty

theory. Thus, experiments are conducted for the use of three gate-oxide thicknesses. In other words, the RTL library contains resources made of three different gate-oxide thicknesses. In this scenario for different benchmark circuits, the maximum reduction was improved in the range of 3–7%, and the average reduction was improved by 2–5%. But, there is increase in the average time penalty for different benchmark circuits, which on an average is 5–11%. This is observed consistently for both 65 nm and 45 nm technologies.

9.8 Conclusions and future directions of research

RTL optimization during HLS has been a topic of research for the last two decades. Based on different technology trends, various objectives including silicon area, circuit performance, switching activity power, subthreshold leakage, and gate-oxide leakage have been considered for optimization [31, 30]. In the current chapter, gate-oxide leakage which is a significant portion of total power dissipation of sub-65 nm technology circuits and contributes to an appreciable portion of total power consumption of CMOS nanometer circuits was targeted. A novel technique which utilizes functional units of multiple oxide thicknesses (multi-T_{ox}) as an attractive option for overall gate-oxide leakage reduction of a datapath circuit has been presented in the current chapter. Multi-T_{ox} based designs may need more masks for the lithographic process of circuit fabrication. However, it is believed that such costs would be compensated by the reduction of energy or power costs. We also present a comparative view of 65 nm and 45 nm technologies. While multi-T_{ox} is highly effective, use of multiple dielectrics using high-κ dielectric materials along with multi-T_{ox} will be explored. The resource selection is being made during scheduling, and we are in the process of evaluating its impact on the area and total power. A heuristic based approach is presented here for functional unit assignment. The use of better optimization techniques can further

improve the results. Further improvement can be achieved by using methods that accurately estimate the logic values. The proposed multi-T_{ox} approach can be used along with multi-V_{dd} and multi-V_{Th} approaches to provide a solution for total power dissipation of CMOS circuits. A comparison of dual-T_{ox} with dual-κ suggests that dual-T_{ox} can be effective but fabrication cost wise better [28, 43]. A comparison of dual-T_{ox} with dual-V_{Th} suggests that dual-T_{ox} can be more effective to reduce both gate-oxide and subthreshold leakage while being less susceptible to process variations [36].

While HLS has been around for a few decades, it is much more relevant today than ever before due to the emergence of sophisticated manufacturing technologies, complex designs, and shorter time to market. The need for HLS methods to capture emerging challenges is ever demanding. For an example in the digital integrated circuit area, security challenges such as hardware-trojan detection during HLS phases are receiving attention [13, 57]. Behavioral synthesis or HLS for analog and mixed-signal circuits from their VHDL-AMS and Verilog-AMS descriptions has been an area of active research for the last several years as well [29, 74]. Research in HLS for micro-electro-mechanical systems (MEMS) or nano-electro-mechanical systems (NEMS) design is also in full swing [12]. These multidirectional research trends are expected to continue for HLS for decades to come. HLS will play its important role of higher level design exploration, and corresponding design decisions before the design proceeds to detailed and effort-intensive phases of lower levels of design abstractions.

Acknowledgments

This chapter is based on previous conference presentations from the authors, such as the following: [32, 35, 45]. The authors would like to acknowledge their graduate students at the University of North Texas (UNT) who helped with preliminary versions of this research.

References

[1] Semiconductor Industry Association, International Technology Roadmap for Semiconductors.http://public.itrs.net

[2] Abe, S.Y., Yanagisawa, M., Togawa, N.: “An Energy-Efficient High-Level Synthesis Algorithm for Huddle-Based Distributed-Register Architectures”. *In: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 576–579 (2012)

[3] Ausin, A.J.B.B., Meindle, J.D.: “Minimum Supply Voltage for Bulk Si CMOS GSI”. *In: Proceedings of International Symposium on Low Power Electronic Design*, pp. 100–102 (1998)

[4] Benini, L., Bogliolo, A., Micheli, G.D.: “A Survey of Design Techniques for System-Level Dynamic Power Management”. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **8**(3), 299–316 (2000)

[5] Bhavnagarwala, A.J., Austin, B.L., Bowman, K.A., Meindl, J.D.: "A Minimum Total Power Methodology for Projecting Limits of CMOS GSI". *IEEE Transactions on VLSI Systems* **8**(3), 235–251 (2000)

[6] Bohr, M.T., Chau, R.S., Ghani, T., Mistry, K.: "The High-κ Solution". *IEEE Spectrum* **44**(10), 29–35 (2007)

[7] Bowman, K.A., Austin, B.L., Eble, J.C., Tang, X., Meindl, J.D.: "A Physical Alpha-Power Law MOSFET Model". *IEEE Journal of Solid-State Circuits* **34**(10), 1410–1414 (1999)

[8] Bowman, K.A., Wang, L., Tang, X., Meindl, J.D.: "A Circuit-Level Perspective of the Optimum Gate Oxide Thickness". *IEEE Transactions on Electron Devices* **48**(8), 1800–1810 (2001)

[9] Cao, Y., Sato, T., Sylvester, D., Orshansky, M., Hu, C.: "New Paradigm of Predictive MOSFET and Interconnect Modeling for Early Circuit Design". *In: Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 201–204 (2000)

[10] Chandrakasan, A., Bowhill, W., Fox, F.: "Design of High-Performance Microprocessor Circuits". *IEEE Press*, Hoboken, NJ, (2001)

[11] Choi, C.H., Oh, K.H., Goo, J.S., Yu, Z., Dutton, W.W.: "Direct Tunneling Current Model for Circuit Simulation". *In: Proceedings of International Electron Devices Meeting* (1999)

[12] Cobb, C.L., Zhang, Y., Agogino, A., Mangold, J.: "Knowledge-Based Evolutionary Linkage in MEMS Design Synthesis". *In: Y.P. Chen, M.H. Lim (eds.) Linkage in Evolutionary Computation*, *Studies in Computational Intelligence*, vol. 157, pp. 461–483. Springer Berlin Heidelberg (2008). DOI 10.1007/978-3-540-85068-7_19. URL http://dx.doi.org/10.1007/978-3-540-85068-7_19

[13] Cui, X., Ma, K., Shi, L., Wu, K.: "High-Level Synthesis for Run-Time Hardware Trojan Detection and Recovery". *In: Proceedings of the 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6 (2014)

[14] Depas, M., Vermeire, B., Mertens, P.W., Meirhaeghe, R.L.V., Heyns, M.M.: "Determination of Tunneling Parameters in Ultra-Thin Oxide Layer Poly-Si/SiO_2/Si Structures". *Elsevier Solid-State Electronics Journal* **38**(8), 1465–1471 (1995)

[15] Garverick, S.L., Sodini, C.G.: "A Simple Model for Scaled MOS Transistor that Includes Field-Dependent Mobility". *IEEE Journal of Solid-State Circuits* **22**(1), 111–114 (1987)

[16] Ghai, D., Mohanty, S., Thakral, G.: "Comparative Analysis of Double Gate FinFET Configurations for Analog Circuit Design". *In: Proceedings of the IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 809–812 (2013)

[17] Gopalakrishnan, C., Katkoori, S.: "Knapbind: An Area-Efficient Binding Algorithm for Low-Leakage Datapaths". *In: Proceedings of 21st International Conference on Computer Design*, pp. 430–435 (2003)

[18] Gopalakrishnan, C., Katkoori, S.: "Resource Allocation and Binding Approach for Low Leakage Power". *In: Proceedings of 16th International Conference on VLSI Design*, pp. 297–302 (2003)

[19] Joshi, S., Kougianos, E., Mohanty, S.P.: "Simscape Based Ultra-Fast Design Exploration of Graphene Nanoelectronic Systems". *In: Proceedings of the 14th IEEE Computer Society Annual Symposium on VLSI (ISVLSI)* (2015)

[20] Khouri, K.S., Jha, N.K.: "Leakage Power Analysis and Reduction during Behavioral Synthesis". *In: Proceedings of International Conference on Computer Design*, pp. 561–564 (2000)

[21] Khouri, K.S., Jha, N.K.: "Leakage Power Analysis and Reduction during Behavioral Synthesis". *IEEE Transactions on VLSI Systems* **10**(6), 876–885 (2002)

[22] Kim, N.S., Austin, T., Blaauw, D., *et al.*: "Leakage Current – Moore's Law Meets Static Power". *IEEE Computer*, pp. 68–75 (2003)

[23] Kougianos, E., Mohanty, S.P.: "A Nature-Inspired Firefly Algorithm Based Approach for Nanoscale Leakage Optimal RTL Structure". *Elsevier The VLSI Integration Journal* **51**, 46–60 (2015)

[24] Lee, D., Blaauw, D.: "Static Leakage Reduction Through Simultaneous Threshold Voltage and State Assignment". *In: Proceedings of the Design Automation Conference*, pp. 191–194 (2003)

[25] Liu, R., Chen, S., Yoshimura, T.: "Post-Scheduling Frequency Assignment for Energy-Efficient High-Level Synthesis". *In: Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 588–591 (2010)

[26] Manzak, A., Chakrabarti, C.: "A Low Power Scheduling Scheme with Resources Operating at Multiple Voltages". *IEEE Transactions on VLSI Systems* **10**(1), 6–14 (2002)

[27] Mohanty, S.P., Ranganathan, N., Krishna, V.: "Datapath Scheduling using Dynamic Frequency Clocking". *In: Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pp. 58–63 (2002)

[28] Mohanty, S.P.: "ILP Based Gate Leakage Optimization using DKCMOS Library during RTL Synthesis". *In: Proceedings of the 9th International Symposium on Quality of Electronic Design (ISQED)*, pp. 174–177 (2008)

[29] Mohanty, S.P.: "Nanoelectronic Mixed-Signal System Design". *McGraw-Hill Education*, New York City, NY, (2015)

[30] Mohanty, S.P.: "Energy and Transient Power Minimization during Behavioral Synthesis". Ph.D. thesis, Department of Computer Science and Engineering, University of South Florida, Tampa, FL, USA (Fall, 2003)

[31] Mohanty, S.P., Gomathisankaran, M., Kougianos, E.: "Variability-Aware Architecture Level Optimization Techniques for Robust Nanoscale Chip Design". *Elsevier Computers & Electrical Engineering* **40**(1), 168–193 (2014)

[32] Mohanty, S.P., Kougianos, E.: "Modeling and Reduction of Gate Leakage during Behavioral Synthesis of NanoCMOS Circuits". *In: Proceedings of the 19th International Conference on VLSI Design*, pp. 83–88 (2006)

[33] Mohanty, S.P., Kougianos, E., Mahapatra, R.N.: "A Comparative Analysis of Gate Leakage and Performance of High-κ Nanoscale CMOS Logic Gates". *In: Proceedings of the 16th ACM/IEEE International Workshop on Logic and Synthesis (IWLS)*, pp. 31–38 (2007)

[34] Mohanty, S.P., Kougianos, E., Pradhan, D.K.: "Simultaneous Scheduling and Binding for Low Gate Leakage Nano-Complementary Metaloxide-Semiconductor Data Path Circuit Behavioural Synthesis". *IET Computers & Digital Techniques* **2**(2), 118–131 (2008)

[35] Mohanty, S.P., Mukherjee, V., Velagapudi, R.: "Analytical Modeling and Reduction of Direct Tunneling Current during Behavioral Synthesis of Nanometer CMOS Circuits". *In: Proceedings of the 14th ACM/IEEE International Workshop on Logic and Synthesis*, pp. 249–256 (2005)

[36] Mohanty, S.P., Panigrahi, B.K.: "ILP Based Leakage Optimization during Nano-CMOS RTL Synthesis: A DOXCMOS versus DTCMOS Perspective". *In: Proceedings of the International Symposium on Biologically Inspired Computing and Applications (BICA)*, pp. 1367–1372 (2009)

[37] Mohanty, S.P., Pradhan, D.K.: "Tabu Search Based Gate Leakage Optimization using DKCMOS Library in Architecture Synthesis". *In: Proceedings of the 12th International Conference on Information Technology (ICIT)*, pp. 3–9 (2009)

[38] Mohanty, S.P., Ranganathan, N.: "A Framework for Energy and Transient Power Reduction during Behavioral Synthesis". *IEEE Transactions on VLSI Systems* **12**(6), 562–572 (2004)

[39] Mohanty, S.P., Ranganathan, N.: "Energy Efficient Datapath Scheduling using Multiple Voltages and Dynamic Clocking". *ACM Transactions on Design Automation of Electronic Systems (TODAES)* **10**(2), 330–353 (2005)

[40] Mohanty, S.P., Ranganathan, N.: "Simultaneous Peak and Average Power Minimization during Datapath Scheduling". *IEEE Transactions on Circuits and Systems I: Regular Papers* **52**(6), 1157–1165 (2005)

[41] Mohanty, S.P., Ranganathan, N., Chappidi, S.K.: "Peak Power Minimization through Datapath Scheduling". *In: Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pp. 121–126 (2003)

[42] Mohanty, S.P., Ranganathan, N., Krishna, V.: "Datapath Scheduling using Dynamic Frequency Clocking". *In: Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pp. 65–70 (2002)

[43] Mohanty, S.P., Velagapudi, R., Kougianos, E.: "Dual-κ versus Dual-T_{ox} Technique for Gate Leakage Reduction: A Comparative Perspective". *In: Proceedings of the 7th International Symposium on Quality Electronic Design*, pp. 564–569 (2006)

[44] Mohanty, S.P., Velagapudi, R., Kougianos, E.: "Physical-Aware Simulated Annealing Optimization of Gate Leakage in Nanoscale Datapath Circuits". *In: Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 1191–1196 (2006)

[45] Mohanty, S.P., Velagapudi, R., Mukherjee, V., Li, H.: "Reduction of Direct Tunneling Power Dissipation during Behavioral Synthesis of Nanometer CMOS Circuits". *In: Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pp. 248–249 (2005)

[46] Mudge, T.N.: "Power: A First Class Design Constraint for Future Architecture and Automation". *In: Proceedings of the International Conference on High Performance Computing*, pp. 215–224 (2000)

[47] Mukherjee, V., Mohanty, S.P., Kougianos, E.: "A Dual Dielectric Approach for Performance Aware Gate Tunneling Reduction in Combinational Circuits". *In: Proceedings of the 23rd IEEE International Conference of Computer Design (ICCD)* (2005)

[48] Nagel, L.W., McAndrew, C.C.: "Is SPICE Good Enough for Tomorrow's Analog?" *In: Proceedings of the IEEE Bipolar/BiCMOS Circuits and Technology Meeting (BCTM)*, pp. 106–112 (2010)

[49] Narendra, S., Keshavarzi, A., Bloechel, B.A., Borkar, S., De, V.: "Forward Body Bias for Microprocessors in 130-nm Technology Generation and Beyond". *IEEE Journal of Solid-State Circuits* **38**(5), 696–701 (2003)

[50] Paik, S., Shin, I., Kim, T., Shin, Y.: "HLS-l: A High-Level Synthesis Framework for Latch-Based Architectures". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **29**(5), 657–670 (2010)

[51] Pant, P., Roy, R.K., Chattejee, A.: "Dual-Threshold Voltage Assignment with Transistor Sizing for Low Power CMOS Circuits". *IEEE Transactions on VLSI Systems* **9**(2), 390–394 (2001)

[52] Rao, R., Srivastava, A., Blaauw, D., Sylvester, D.: "Statistical Analysis of Subthreshold Leakage Current for VLSI Circuits". *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **12**(2), 131–139 (2004)

[53] Rao, R.M., Burns, J.L., Brown, R.B.: "Circuit Techniques for Gate and Sub-Threshold Leakage Minimization in Future CMOS Technologies". *In: European Solid-State Circuits Conference*, pp. 313–316 (2003)

[54] Roy, K., Krishnammthy, R.: "Design of Low Voltage CMOS Circuits: Tutorial Guide". *In: Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 3.2.1–3.2.29 (2001)

[55] Roy, K., Mukhopadhyay, S., Meimand, H.M.: "Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits". *Proceedings of the IEEE* **91**(2), 305–327 (2003)

[56] Sakurai, T., Newton, A.R.: "Alpha-Power Law MOSFET Model and Its Applications to CMOS Inverter Delay and Other Formulas". *IEEE Journal of Solid-State Circuits* **25**(2), 584–594 (1990)

[57] Sengupta, A., Bhadauria, S.: "Untrusted Third Party Digital IP Cores: Power-Delay Trade-Off Driven Exploration of Hardware Trojan Secured Datapath during High Level Synthesis". *In: Proceedings of the 25th IEEE/ACM Great Lake Symposium on VLSI (GLSVLSI)* (2015)

[58] Sengupta, A., Sedaghat, R.: "Integrated Scheduling, Allocation and Binding in High Level Synthesis using Multi Structure Genetic Algorithm Based Design Space Exploration". *In: Proceedings of the 12th International Symposium on Quality Electronic Design (ISQED)*, pp. 1–9 (2011)

[59] Shiue, W.T., Chakrabarti, C.: "Low-Power Scheduling with Resources Operating at Multiple Voltages". *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing* **47**(6), 536–543 (2000)

[60] Sill, F., Grassert, F., Timmermann, D.: "Total Leakage Power Optimization with Improved Mixed Gates". *In: Proceeding of the 18th Symposium on Integrated Circuits and Systems Design*, pp. 154–159 (2005). DOI 10.1109/SBCCI.2005.4286849

[61] Simunic, T., Benini, L., Acquaviva, A., Glynn, P., Micheli, G.D.: "Dynamic Voltage Scaling and Power Management for Portable Systems". *In: Proceedings of the ACM/IEEE Design Automation Conference*, pp. 524–529 (2001)

[62] Singh, D., Rabaey, J.M., Pedram, M., *et al.*: "Power Conscious CAD Tools and Methodologies: A Perspective". *In: Proceedings of the IEEE* **83**(4), 570–594 (1995)

[63] Sirisantana, N., Roy, K.: "Low-Power Design using Multiple Channel Lengths and Oxide Thicknesses". *IEEE Design & Test of Computers* **21**(1), 56–63 (2004)

[64] Sirisantana, N., Wei, L., Roy, K.: "High-Performance Low-Power CMOS Circuits using Multiple Channel Length and Multiple Oxide Thickness". *In: Proceedings of the IEEE International Conference on Computer Design*, pp. 227–232 (2000)

[65] Sultania, A.K., Sylvester, D., Sapatnekar, S.S.: "Tradeoffs between Gate Oxide Leakage and Delay for Dual T_{ox} Circuits". *In: Proceedings of Design Automation Conference*, pp. 761–766 (2004)

[66] Sylvester, D., Kaul, H.: "Power-Driven Challenges in Nanometer Design". *IEEE Design and Test of Computers* **13**(6), 12–21 (2001)

[67] Sze, S.M.: "Physics of Semiconductor Devices". John Wiley, Hoboken, NJ, (1981)

[68] Sze, S.M.: "Semiconductor Devices: Physics and Technology". John Willey, Hoboken, NJ, (2002)

[69] Vogel, E.M., Ahmed, K.Z., Hornung, B., *et al.*: "Modeled Tunnel Currents for High Dielectric Constant Dielectrics". *IEEE Transactions on Electron Devices* **45**(6), 1350–1355 (1998)

[70] Wang, F., Wu, X., Xie, Y.: "Variability-Driven Module Selection with Joint Design Time Optimization and Post-Silicon Tuning". *In: Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 2–9 (2008)

[71] Weste, N.H.E., Harris, D.: "CMOS VLSI Design: A Circuit and Systems Perspective". Addison Wesley, Boston, MA, (2005)

[72] Yao, J., Agrawal, V.D.: "Dual-Threshold Design of Sub-Threshold Circuits". *In: Proceedings of the IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*, pp. 1–2 (2013)

[73] Yu, B., Ju, D.H., Lee, W.C., *et al.*: "Gate Engineering for Deep-Submicron CMOS Transistors". *IEEE Transactions on Electron Devices* **45**(6), 1253–1262 (1998)

[74] Zeng, K., Huss, S.A.: "RAMS: A VHDL-AMS Code Refactoring Tool Supporting High Level Analog Synthesis". *In: Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pp. 266–267 (2005)

[75] Zhao, Z., Bian, J., Liu, Z., Wang, Y., Zhao, K.: "High Level Synthesis with Multiple Supply Voltages for Energy and Combined Peak Power Minimization". *In: Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems*, pp. 864–867 (2006)

Chapter 10

Green on-chip inductors for three-dimensional integrated circuits: concepts, algorithms and applications

Umamaheswara Rao Tida[1], *Cheng Zhuo*[2] *and Yiyu Shi*[1]

This chapter deals with a completely different aspect of circuit and design as compared to the previous chapters. This chapter discusses three-dimensional integrated circuits (3D ICs) as compared to the planner integrated circuits of the previous chapters. The 3D ICs have shown significant promise for the future post-CMOS era circuits and systems to build high-performance systems with minimal silicon foot print. This chapter specifically discusses practical approaches to through-silicon-via (TSV) inductors which constitute the vertical signal, power and thermal paths which is very critical for 3D ICs.

10.1 Introduction

3D ICs are generally considered to be the most promising alternative that offers a path beyond Moore's Law. Instead of making transistors smaller, it makes use of the vertical dimension for higher integration density, shorter wire length, smaller footprint, higher speed and lower power consumption and is fully compatible with current technology [11].

The TSV is a critical enabling technique for 3D ICs, which forms vertical signal, power and thermal paths. While many challenges still exist in 3D ICs, a big one is related to TSVs: they are large in size, typically 5–10× larger than the standard cells in 32 nm process [12]. Yet their diameters do not scale with the devices due to imposed limitations of wafer handling and aspect ratios (ARs). International Technology Roadmap for Semiconductors (ITRS) suggests that the TSV diameter will remain almost constant in 2012–2015. On the other hand, a large number of TSVs are needed to deliver signal and power, to dissipate heat and to provide redundancy. Moreover,

[1]Missouri University of Science and Technology, MO 65409, USA
[2]Intel Corporation, Hillsboro

to guarantee high yield rate, foundries typically impose a minimum TSV density rule to maintain the planarity of the wafer during chemical and mechanical polishing. For example, Tezzaron requires at least one TSV in every 250 μm $\times$ 250 μm area [5]. Similar rules are imposed by Intel, TSMC and other 3D foundries. To satisfy this rule, lots of dummy TSVs need to be inserted, which further increase the area overhead.

To alleviate the problem, there have been efforts in the literature to make use of those dummy TSVs for alternative purposes. In this chapter, we are particularly interested in the application of TSVs toward on-chip inductors, which are the critical components in various microelectronic applications, e.g., on-chip voltage regulators, resonant clocking, voltage control oscillators, power amplifiers and radio frequency (RF) circuits. The design rules for the co-existence of TSV inductors and normal digital/analog/RF components are the same as those for regular 3D designs, as we are not employing any dedicated process steps.

Conventional implementation of on-chip inductors uses multi-turn planar spiral structure. This structure occupies a significant area and requires special RF process for higher quality factor. For example, Bian *et al.* [26] reported an inductor which occupies 78,400 μm^2 routing area, equivalent to the area of 62K gates in 45 nm technology. In 3D ICs, however, it is possible to utilize TSVs to build vertical inductors [2, 8, 10, 22–25, 30]. One example of a toroidal TSV inductor in a two-tier 3D IC is shown in Figure 10.1. An apparent advantage of such TSV inductors is the minimal footprint on routing layers and accordingly high inductance density. However, since it is completely buried in the lossy substrate, its quality factor is inferior compared

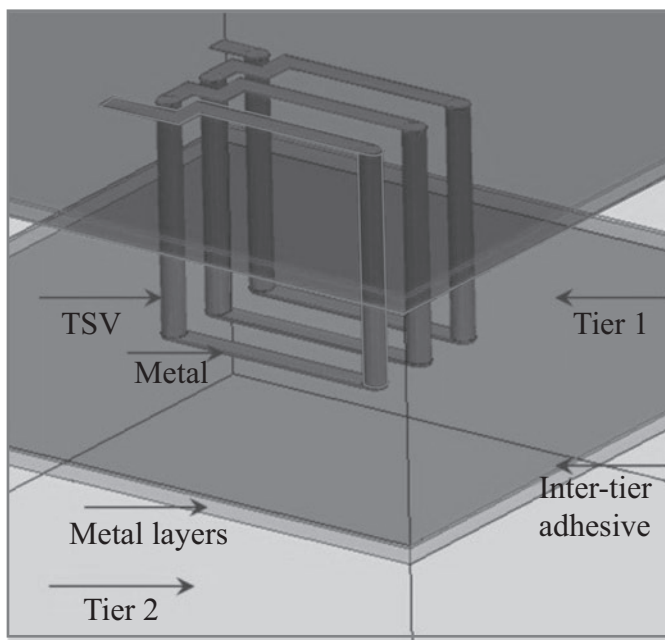


Figure 10.1 Structure of TSV inductor

with that of the 2D spiral inductor. Accordingly, as pointed out in Reference 30, TSV inductor can be used when area is the only concern. This essentially declares that such a TSV inductor is useless in practice.

To make TSV inductors practical, two fundamental questions need to be understood:

- First, what are the parameters that can effectively improve the TSV inductor performance?
- Second, is there any shield mechanism that can be used to reduce substrate loss if the ac losses are dominant?

This chapter provides answers to both questions. We first observe the effect of various parameters on the performance of the inductor.

We then focus on two low-frequency applications utilizing TSV inductors for low-frequency applications, i.e., inductive DC–DC converters and LC resonant clocking. Inductive DC–DC converters become prominent for on-chip voltage conversion because of their high efficiency compared with other types of converters (e.g., linear and capacitive converters). On the other hand, to reduce on-chip power, LC resonant clocking has become an attracting option due to its same amplitude and phases compared to other resonant clocking methods such as standing wave [7] and rotary wave [15]. A major challenge for both applications is associated with the inductor area required. We then put forward a novel shield technique using the micro-channel, which has been used in 3D IC industry including IBM and Nanonexus as a low-cost cooling technique [6], to reduce the substrate loss. Shielding changes the TSV inductor concept from just a fantasy to something practical for high-frequency applications.

The organization of this chapter is as follows. Section 10.2 describes the effect of process and design parameters on the performance of an inductor. Low-frequency applications using these inductors are discussed in Section 10.3. Section 10.4 explains the micro-channel shielding technique and shows its effect on the performance of inductor for high frequencies, and conclusions are given in Section 10.5.

10.2 Effect of various parameters of an on-chip inductor

The general structure of existing toroidal TSV inductors is shown in Figure 10.1, which is composed of front/back metals and TSVs in a toroidal structure for face-to-back bonding. The most attractive advantage of such a TSV inductor is its minimal footprint on the silicon surface. In addition, no patterned ground shield (PGS) is necessary as the majority of the magnetic flux run in parallel with metal wires (in the horizontal plane).

In this section, we will study how various process and design parameters affect the inductance, the quality factor, as well as the self-resonant frequency (SRF) of the TSV inductor. All the simulations in this chapter are done using ANSYS full-wave simulator HFSS with mixed order basis function. Our machine is a 64-bit Dell T7500 Windows server with 2.4 GHz duo-core Xeon CPU and 96 GB memory. For clarity

Table 10.1 List of parameters, the respective default unit and ranges of interest

Type	Notation	Meaning	Range
Process	H (μm)	Substrate height	30–120
	σ (S/m)	Substrate conductivity	0–10,000
	D (μm)	TSV diameter	2–15
	d (μm)	Liner thickness	0.2–0.7
	h (μm)	Metal height	0.2–3
Design	N	Number of turns	2–6
	T	Number of tiers	2–4
	P (μm)	Loop pitch	13–23
	W (μm)	Width of metal strip	3–12
	f (GHz)	Operating frequency	0.15, 1, 5, 10

purposes, we outline the parameters of study in Table 10.1. The practical range of interest for each parameter is also listed.

There are four things worthwhile to note here:

1. We used the inductor designs of up to four tiers (according to Reference 32, 3D ICs of up to five tiers have already been fabricated). Since the bottom tier does not need any TSV, the actual inductor is formed in the top $T - 1$ tiers.
2. To achieve maximum quality factor, the cross-sectional area should be square. In other words, once we fix the number of tiers T, the TSV pitch should be $(T - 1)H$, where H is the height of a single tier.
3. The substrate height and the TSV diameter are chosen such that the TSV AR is between 5:1 and 20:1, in accordance with ITRS.
4. The 150 MHz operating frequency represents applications such as on-chip voltage regulator applications, while 1/5/10 GHz represents resonant clocking or RF applications.

To study the impact of various parameters, we use the control variable method to change one parameter at a time. The nominal settings are illustrated in Figure 10.2: Process parameters: $H = 60\,\mu$m, $\sigma = 10$ S/m, $D = 6\,\mu$m, $d = 0.2\,\mu$m. Design parameters: $N = 1$, $T = 2$, $P = 18\,\mu$m (not shown), $W = 6\,\mu$m. In addition to these parameters of study, for each tier, we assume a normal process with eight metal layers. The metal layers have a total thickness (including field dioxide) of 4 μm. The metal strips connecting TSVs are implemented using M1 (0.3 μm thick) and backside metal (0.8 μm thick). The corresponding inductance and quality factor vs. frequency plot for the above nominal settings are shown in Figure 10.3. From the figure, we can see that the TSV inductor performs better than the spiral inductor for a particular range of frequency and then the performance degrades substantially. For our low-frequency applications, the TSV inductor performs on par or better than the spiral inductor, hence no extra shielding is required.

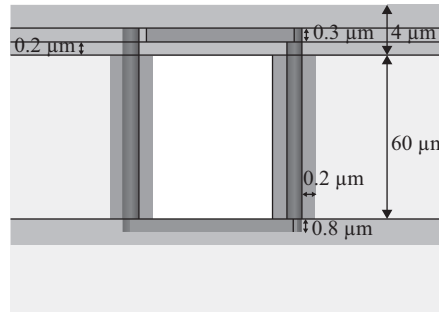


Figure 10.2 Nominal settings of an inductor

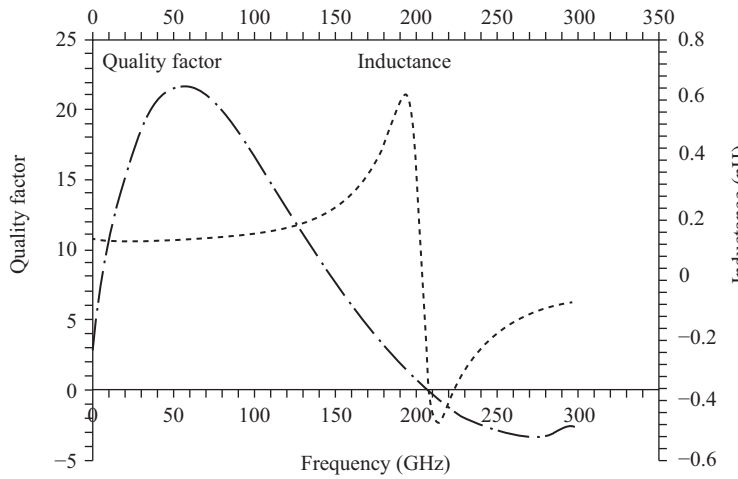


Figure 10.3 Quality factor and inductance vs. frequency for the TSV inductor with nominal settings

10.2.1 Impact of process parameters

In this section, we study the impact of process parameters on the inductance, quality factor and SRF of the TSV inductor, hoping to suggest directions for dedicated 3D TSV inductor process development in the near future.

10.2.1.1 Substrate height (H)

The quality factor and the inductance for different substrate heights and operating frequencies are shown in Figure 10.4. Based on the analogy to the spiral inductors, the inductance should be proportional to $Hln(H)$. This can be clearly verified by curve fitting. In terms of quality factor, it increases with H, but at different rates for different frequencies. Finally, although not shown in the figure, we note that SRF decreases from over 250 GHz to 100 GHz when H increases from 30 µm to 120 µm.

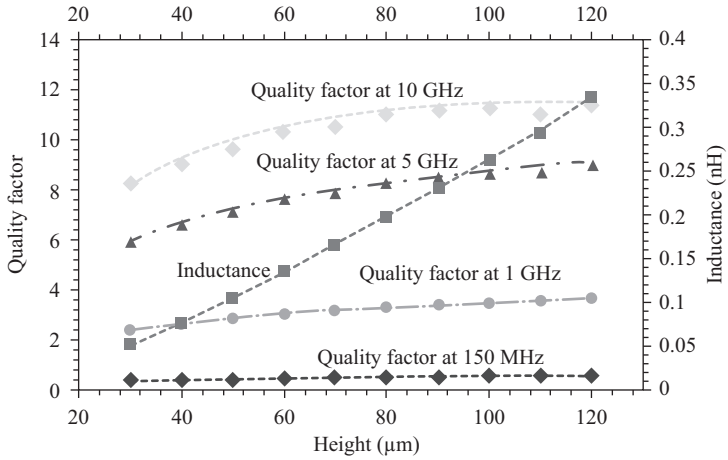


Figure 10.4 Quality factor and inductance vs. substrate height

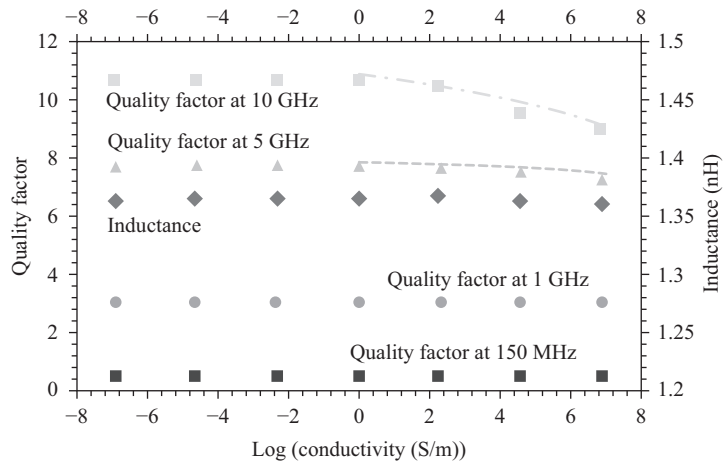


Figure 10.5 Q and L vs. substrate conductivity σ (Note that log-scale is used for the x-axis)

Observation 1: For the range of interest, increasing substrate height increases both the inductance and the quality factor, but reduces the SRF.

10.2.1.2 Substrate conductivity (σ)

The quality factor and the inductance for different substrate conductivities and operating frequencies are shown in Figure 10.5. From the figure, we can see that the inductance is not directly impacted by σ ($L = 0.13$ nH). On the other hand, when σ is low (corresponding to the lightly doped substrate) or when the frequency is

low (150 MHz or 1 GHz), the quality factor almost remains constant, because in this region the quality factor loss is mainly due to the ohmic loss in the inductor. When both σ and the frequency are high, the quality factor decreases with σ, at higher rate for higher frequencies. This is due to the fact that in this region, the loss mainly occurs in the substrate. Finally, although not shown in the figure, we note that SRF decreases from over 200 GHz to 60 GHz when σ increases from 0 S/m to 10,000 S/m.

Observation 2: For low substrate doping density ($\sigma < 10$ S/m) or low frequency (<1 GHz), the ohmic loss of the inductor dominates. When the doping density is high and the frequency is high, the substrate loss dominates.

Observation 3: For the range of interest, increasing substrate conductivity does not change the inductance and has little impact on the quality factor at low frequency or low substrate conductivity. It reduces the quality factor gradually at high frequency for high substrate conductivity. The SRF drops with the increase of substrate conductivity.

10.2.1.3 TSV diameter (*D*)

The quality factor and the inductance for different TSV diameters and operating frequencies are shown in Figure 10.6. Based on the analogy to the spiral inductors (metal width), the inductance should be proportional to $ln(H/D)$.

In terms of quality factor, the quality factor should increase with D as the resistance of the inductor becomes smaller. Apparently, at higher frequency, the quality factor is larger and the slope w.r.t. D is higher. The larger slope is due to the effect of further AC resistance reduction from substrate coupling at higher frequencies.

Finally, although not shown in the figure, we note that SRF is almost constant (~200 GHz) for our diameter range (3–15 μm).

Observation 4: For the range of interest, increasing TSV diameter reduces the inductance, increases the quality factor and does not change the SRF significantly.

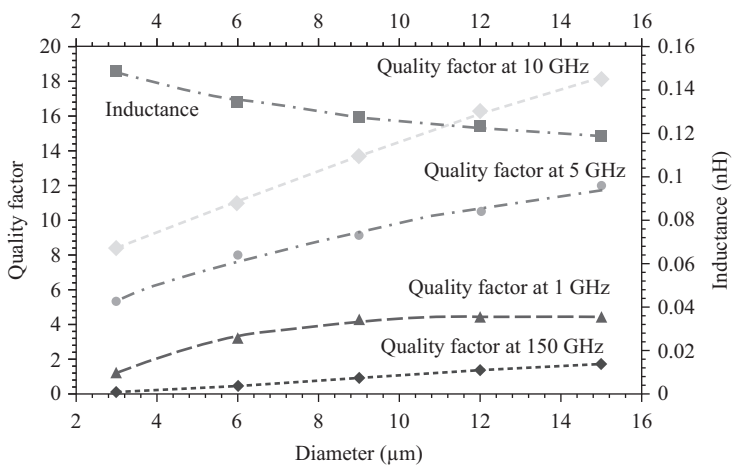


Figure 10.6 Q and L vs. diameter D

10.2.1.4 Liner thickness (*d*)

The quality factor and the inductance for different liner thickness and operating frequencies are shown in Figure 10.7. This parameter is unique to the TSV inductor, and based on the plot, it can be seen that *d* has little impact on the inductance and the quality factor. It also has subtle impact on the SRF.

Observation 5: For the range of interest, TSV liner thickness has subtle impact on the TSV inductor behavior.

10.2.1.5 Metal height (*h*)

The quality factor and the inductance for different metal heights and operating frequencies are shown in Figure 10.8. The inductance decreases as *h* increases due to

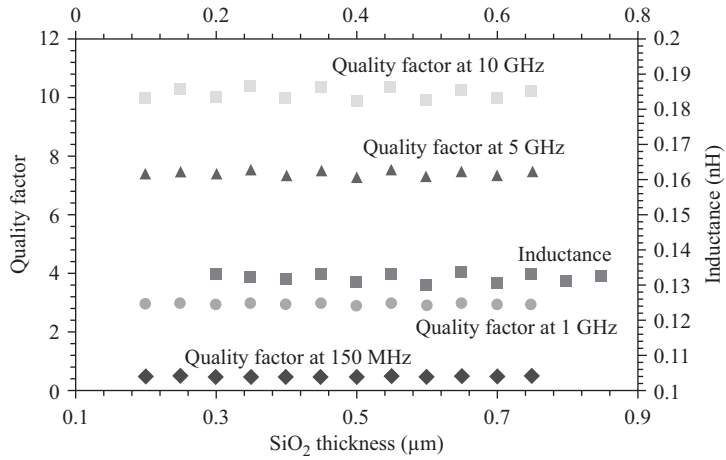


Figure 10.7 Q and L vs. linear thickness d

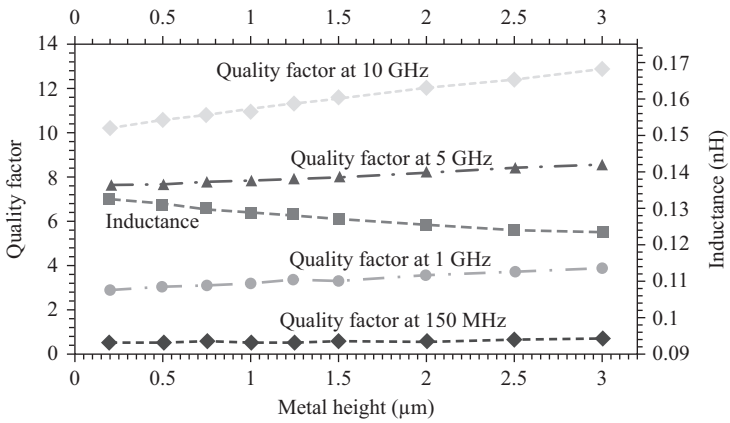


Figure 10.8 Q and L vs. metal height h

the increased capacitive coupling. In terms of quality factor, it increases with h, as the ohmic loss becomes smaller. The impact of h on Q also becomes more profound at higher frequency.

Finally, we note that the SRF remains almost constant (~250 GHz).

Observation 6: For the range of interest, increasing the metal height h decreases the inductance and has little impact on the quality factor at low frequency. It increases the quality factor at high frequency. The SRF does not change with h significantly.

10.2.2 Design parameters

10.2.2.1 Number of turns (N)

The quality factor and the inductance for different number of turns and operating frequencies are shown in Figure 10.9. Based on the analogy to the 2D spiral inductors, inductance should be proportional to N^k.

In terms of quality factor, a few interesting phenomena can be observed.

- First, there exists a particular N_c that gives maximum quality factor.
- Second, such N_c decreases with the frequency. At 150 MHz and 1 GHz, it is over 6 (beyond the scope of the plot), and as a result, the quality factor increases monotonically with N within our range of interest. At 5 GHz, the peak quality factor is reached at $N_c = 3$. At 10 GHz, it drops to 1, and thus the quality factor monotonically decreases with N.
- Third, for higher frequency, the quality factor changes (either increases or decreases) faster with N.

Finally, although not shown in the figure, we note that SRF is decreasing from over 200 GHz to 40 GHz when N increases from 1 to 6.

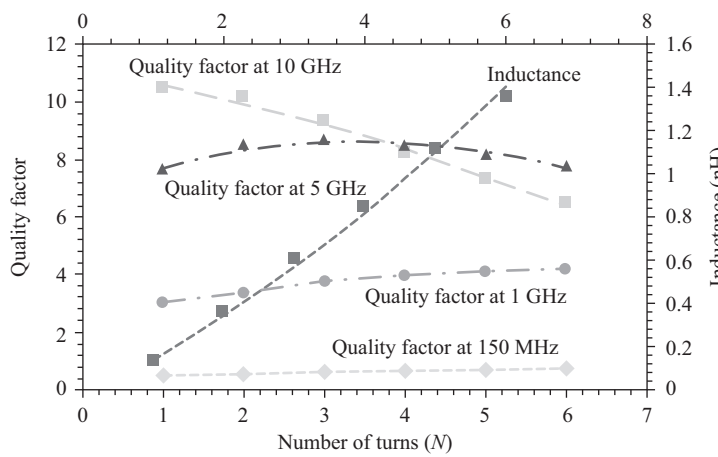


Figure 10.9 Q and L vs. number of turns N

Observation 7: For the range of interest, increasing the number of turns N increases the inductance. There might exist a critical number of turns N_c that gives maximum quality factor, and such N_c decreases with the frequency. The SRF drops rapidly with the increase of N.

10.2.2.2 Number of tiers (T)

The quality factor and the inductance for different number of tiers and operating frequencies are shown in Figure 10.10. The parameter T looks similar to the TSV substrate height H, but it is the non-conducting inter-layer adhesive that makes it different. We can expect that the adhesive layer should have little impact on the inductance, but big impact on the quality factor.

In terms of the quality factor, first, there exists a particular T_c that gives maximum quality factor. Second, such T_c decreases with the frequency. At 1 GHz, it is over 6 and as a result, the quality factor increases monotonically with T. At 5 GHz, the peak quality factor is reached at $T_c = 4$. At 10 GHz, it drops to 3. Third, for higher frequency, the quality factor changes (either increase or decrease) faster with T. Finally, although not shown in the figure, we note that SRF decreases from over 250 GHz to 38 GHz when T increases from 2 to 6.

Observation 8: For the range of interest, increasing the number of tiers T increases the inductance. There might exist a critical number of tiers T_c that gives maximum quality factor, and such T_c decreases with the frequency. The SRF drops rapidly with the increase of T.

Before we continue, one more thing we would like to study is how T_c changes with different N, and how N_c changes with different T, at the same frequency. We again vary N and T based on nominal setting to perform simulation, and the results at 5 GHz are reported in Tables 10.2 and 10.3, respectively. We also include the corresponding Q_{max} at T_c (or N_c). From Table 10.2, it can be seen that with more turns, the number

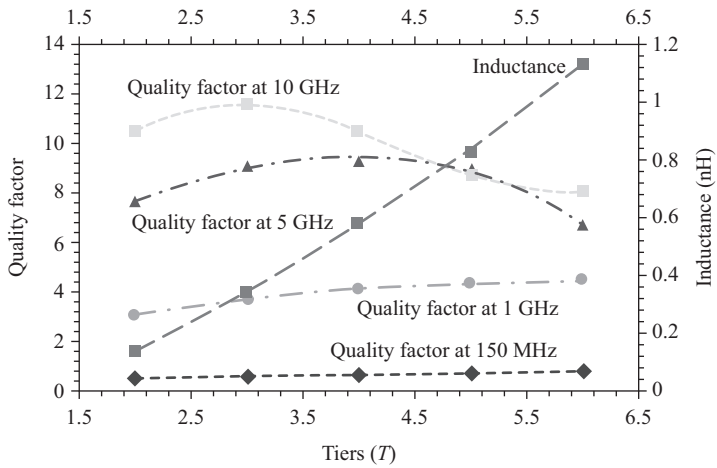


Figure 10.10 Q and L vs. number of tiers T

Table 10.2 T_c and Q vs. N (measured at 5 GHz)

N	**1**	**2**	**3**	**4**	**5**
T_c	4	2	2	2	2
Q_{max}	9.32	8.5	8.77	8.6	8.23

Table 10.3 N_c and Q vs. T (measured at 5 GHz)

T	**2**	**3**	**4**	**5**
N_c	3	1	1	1
Q_{max}	8.77	9.11	9.32	8.95

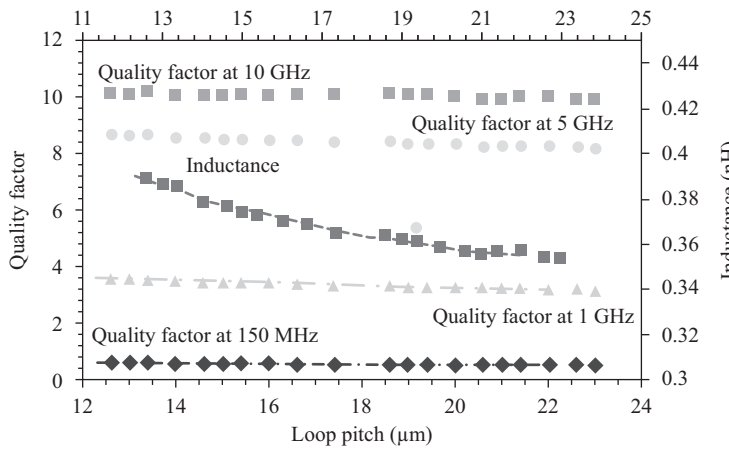


Figure 10.11 Q and L vs. loop pitch P

of tiers that gives maximum quality factor decreases. Similarly, from Table 10.3, with more tiers, the number of turns that gives maximum quality factor decreases.

10.2.2.3 Loop pitch (*P*)

The quality factor and the inductance for different loop pitches and operating frequencies are shown in Figure 10.11. This is a unique parameter for the TSV inductor. If the loop pitch increases, the inductance decreases slightly, mainly due to the reduced magnetic flux. On the other hand, the quality factor decreases with the increase of *P* at lower frequencies and remains almost constant at higher frequencies. This is because at lower frequencies the loss is mainly due to the metal resistance which increases with *P*. At higher frequencies, the substrate loss starts to dominate, which decreases with the magnetic flux (with the increase of *P*). It conforms to our Observation 2.

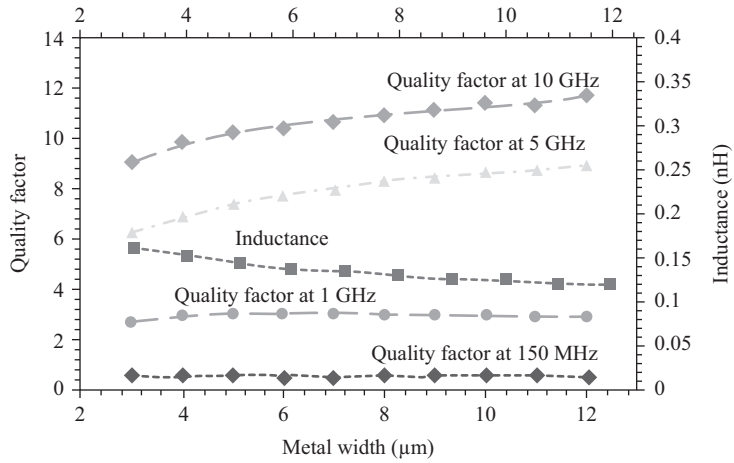


Figure 10.12 Q and L vs. metal width (W)

Finally, we note that the SRF remains almost constant (∼250 GHz).

Observation 9: For the range of interest, increasing the loop pitch P slightly decreases the inductance. The quality factor also slightly decreases with P at low frequency and remains almost constant at high frequency. The SRF does not change significantly with P.

10.2.2.4 Metal width (W)

The quality factor and the inductance for different metal widths and operating frequencies are shown in Figure 10.12. The inductance decreases as W increases due to the increased capacitive coupling.

In terms of quality factor, it increases with W, as the ohmic loss becomes smaller. The impact of W on Q also becomes more profound at higher frequency. Fitting results suggest that the quality factor almost remains constant at 150 MHz. At high frequency, the quality factor increases with cubic trend, which is due to the effect of further AC resistance reduction from substrate coupling.

Finally, we note that the SRF remains almost constant (∼250 GHz).

Observation 10: For the range of interest, increasing the metal width W decreases the inductance and has little impact on the quality factor at low frequency. It increases the quality factor at high frequency. The SRF does not change with W significantly.

10.3 Low-frequency applications

10.3.1 DC–DC converter design

In this section, we will first review the general design of an inductive DC–DC converter in Section 3.1. We will then demonstrate various possible TSV inductor

structures for this application and compare their metrics such as inductance L, quality factor Q, series DC resistance (R_{dc}) and AC resistance (R_{ac}) with a conventional spiral inductor in Section 3.2. Finally, we will compare the performance and area of the inductive DC–DC converters using TSV inductors and conventional spiral inductors in Section 3.3.

10.3.1.1 Overview of inductive DC–DC converters

DC–DC converters are an essential component for integrated systems with multiple power domains. Recently, there has been a groundswell of research interests in implementing DC–DC converters on chip, which isolates the internal system from the large resonant voltage swings due to package parasitics. It also minimizes the external environmental effects.

Compared with other types of DC–DC converters such as linear converters and capacitive converters, inductive converters are known for their high efficiency, which is defined as the ratio of total output power to the total input power (i.e., power delivered by the source) [14].

There are various ways to implement inductive converters such as single-phase buck converters, interleaved buck converters and interleaved buck converters with magnetic coupling [14]. They span the tradeoffs between design complexity and performance, which can be evaluated by the voltage ripple, power efficiency and output droop [14, 29]. In this section, we will use single-phase buck converter as a vehicle to demonstrate the efficacy of TSV inductors in DC–DC converter designs.

The circuit diagram for the single-phase buck converter is shown in Figure 10.13(a). The output voltage ripple occurs due to the charging and discharging of the capacitor and decreases with increase of inductance and capacitance. Despite its simplicity, the output ripple is usually high. In order to reduce the output ripple, the interleaved buck converter with magnetic coupling has been proposed (shown in Figure 10.13(b)).

10.3.1.2 TSV inductor design

In this section, we explore two TSV inductor structures. The first structure (toroidal) is shown in Figure 10.14. To reduce resistance of the inductor, the horizontal metal strips to connect TSVs use M9 of the top tier and the bottom tier. It is easy to find out based on the process parameters discussed previously that the total height of the TSV inductor is 187 μm. As such, in order to maximize the quality factor, the TSV pitch is also set of 165 μm (square cross-sectional area). Furthermore, in order to match the inductance of the conventional spiral inductor discussed above, a total of three turns are used, with a loop pitch of 5 μm. The simulated R_{dc} is 170 mΩ, R_{ac} is 254 mΩ, Quality factor is 8.5, and the inductance is 1.72 nH.

The second structure (vertical spiral) is inspired by the conventional spiral inductor, as shown in Figure 10.15. We implemented a three-turn structure. The outermost loop uses M9 of the top tier and M7 of the bottom tier; the middle loop uses M8 of both tiers; and the innermost loop uses M7 of the top tier and M9 of the bottom tier.

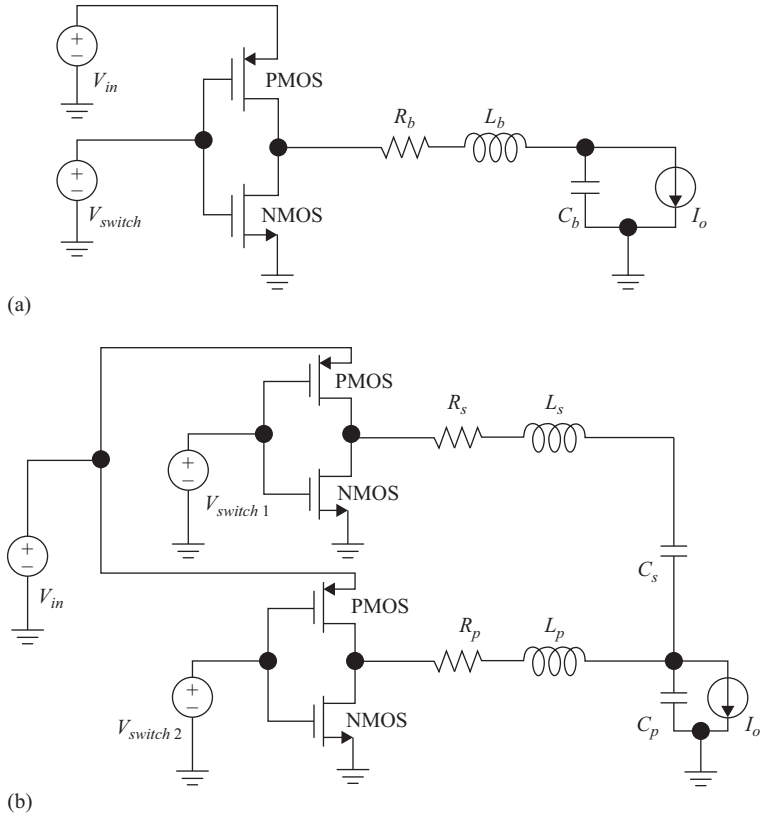
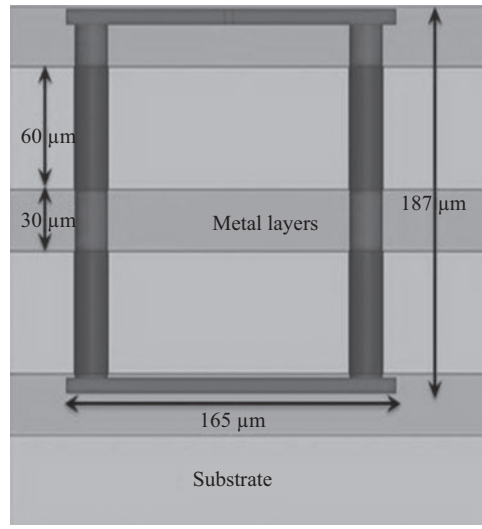


Figure 10.13 (a) Single-phase buck converter and (b) interleaved buck converter with magnetic coupling schematics

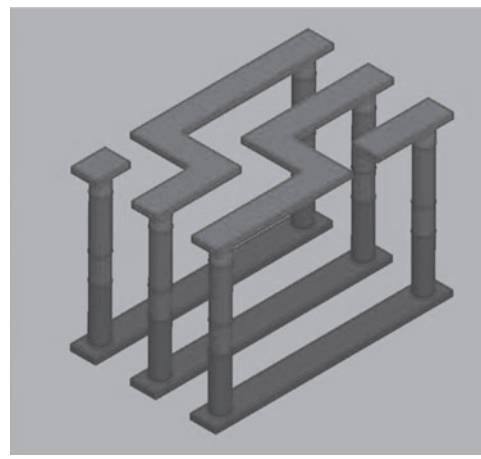
The simulated R_{dc} is 232 mΩ, R_{ac} is 354 mΩ, quality factor is 6.1, and the inductance is 1.73 nH.

For comparison purposes, we also implemented a spiral inductor following the description in Reference 4. For this design, we assumed a process of nine metal layers (M1–M9) with 30 μm thickness in total. The top two metal layers M8 and M9 are of 7 μm thick each. The diameter of the spiral inductor is 336 μm. The metal width is 30 μm, and the pitch is 5 μm. The substrate height is 300 μm (no wafer thinning). The PGS is constructed 5 μm below the spiral inductor(s). The PGS uses 10 μm metal width with 1 μm pitch. The simulated R_{dc} is 178 mΩ, R_{ac} is 404 mΩ, quality factor is 5.4, and the inductance is 1.73 nH.

The above results are summarized in Table 10.4. We also simulated the quality factor Q of both TSV inductors as well as the conventional spiral inductor with respect to frequency. The results with respect to frequency are shown in Figure 10.15. The frequency range is limited to 1 GHz considering the target application of DC–DC converters. From the figure, we can see that the toroidal TSV inductor has the



(a)



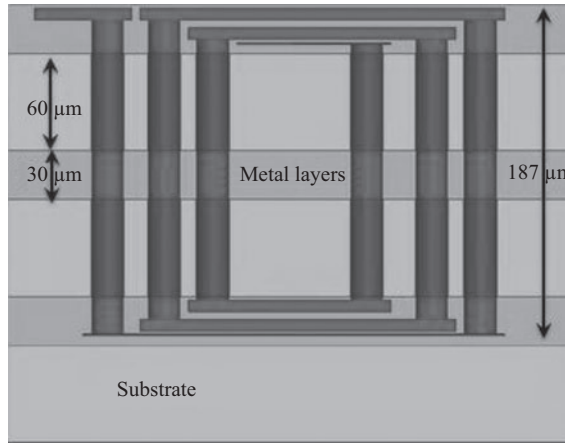
(b)

Figure 10.14 Toroidal TSV inductor. (a) Cross-sectional and (b) rotated view (not to scale) [21]

smallest R_{ac} and the highest Q compared with the other two inductors, due to its larger cross-sectional area.

10.3.1.3 Experimental results

To compare the impact of TSV inductors and conventional spiral inductors on inductive DC–DC converters, we implemented a total of three designs using conventional spiral inductor [16], toroidal TSV inductor and vertical spiral TSV inductor. The target



(a)

(b)

Figure 10.15 Vertical TSV inductor. (a) Cross-sectional and (b) rotated view (not to scale) [21]

design specs are listed in Table 10.5 where the input voltage V_{in} is 1.5 V, desired output voltage V_{out} is 1.2 V, the maximum output droop is 15%, frequency of operation is 200 MHz, and rise/fall times is 50 ps.

For each design, we tune the transistor sizes and the capacitors as shown in Figure 10.13(a) to achieve the design specs. The resulting circuit parameters are shown in Table 10.6 (same for all the three inductors) where duty cycle D is 80, the load capacitance C_s is 10 nF, the series resistance of C_s is 1 mΩ, and the W/L of PMOS and NMOS are 9.6 mm/50 nm and 1.92 mm/50 nm, respectively (Figure 10.16).

Table 10.4 Parameters for different inductor types (200 MHz)

Type	Spiral	Toroidal	Vertical spiral
L (nH)	1.73	1.72	1.73
R_{dc} (mΩ)	178	170	232
R_{ac} (mΩ)	404	254	354
Q	5.4	8.5	6.1

Table 10.5 Target design specs for all DC–DC converters

V_{in}	V_{out}	Max output droop	Frequency	Rise/fall times
1.5 V	1.2 V	15%	200 MHz	50 ps

Table 10.6 Circuit specifications

D (%)	80
C_s	10 nF
ESR of C_s	1 mΩ
PMOS size (W/L)	9.6 mm/50 nm
NMOS size (W/L)	1.92 mm/50 nm

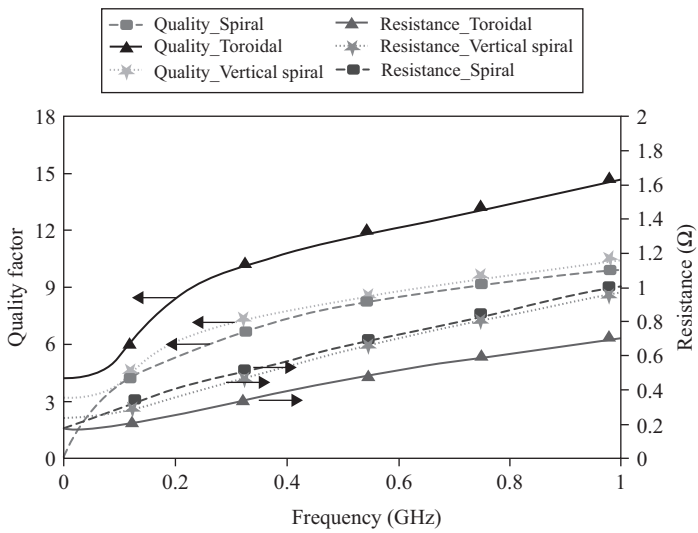


Figure 10.16 Q and L vs. metal width (W)

The peak efficiency, ripple and area comparison for the three designs are shown in Table 10.7. From the table, we can see that they exhibit almost the same efficiency, although the peak efficiency is slightly higher for the design using toroidal TSV inductor and lower for the one using vertical spiral TSV inductor. It is mainly due to the lower R_{dc} for the toroidal TSV inductor and higher R_{dc} for the vertical spiral TSV inductor, as can be seen from Table 10.4. The output voltage ripple for all the cases are also almost the same, as it depends on the inductance and the capacitance values, which are the same in all the cases. The area for all inductors is measured by the total routing resource occupied. For TSV inductors, the area also includes the substrate occupied by the TSVs. Compared with the spiral inductor, the toroidal TSV inductor and the vertical spiral TSV inductor can reduce the area by 3.5× and 4.3×, respectively [21].

We also study how the efficiencies of the three designs change with the load current. As shown in Figure 10.17, all the three designs achieve almost the same efficiency in the range from 0 to 600 mA, which is the maximum load under the output voltage droop limit. The optimal load for all the designs is 400–600 mA. Note that according to the discussion in Reference 30, the maximum current a TSV can handle is around 2 A, so the TSV inductors can be applied without causing any reliability issues.

Table 10.7 Peak efficiency, ripple and area comparison for single-phase Buck converters with different inductor types

Type	**Spiral**	**Toroidal**	**Vertical spiral**
Peak efficiency (%)	76.6	77.1	74
Ripple (mV)	45	45	46
Inductor area (μm^2)	225,792 (1)	64,999 (1/3.5×)	53,120 (1/4.3×)

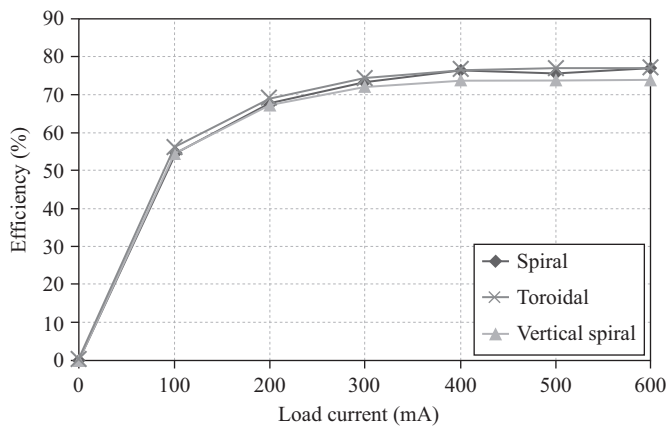


Figure 10.17 Efficiency vs. load current for single-phase buck converters

From all the above results, it is clear that compared with the single-phase buck converter design using conventional spiral inductors, the designs using TSV inductors can achieve the same efficiency and ripple, but with significantly reduced inductor area. In addition, the toroidal TSV inductor is superior over the vertical spiral one in this application.

10.3.2 Resonant clocking implementation

In this section, we will first review the general design of an LC resonant clock distribution network (CDN) in Section 4.1. We will then demonstrate how we use TSV inductors to replace conventional spiral inductors there. Finally, we will compare the performance and area of the LC resonant CDN using TSV inductors and conventional spiral inductors in Section 3.3.

10.3.2.1 Overview of LC resonant clocking

Power consumption has become a major limiting factor to many high-performance designs today. Despite the development of various low power design techniques, the CDN consumes a significant portion (i.e., 30–70%) of the total on-chip power, mainly due to the constant switching and large capacitance loads from the registers.

To reduce CDN power, many prior works have examined techniques such as logic reordering [9], clock and power gating [13] and dynamic voltage and frequency scaling [3]. However, these techniques either require modification of the circuit logic or are not effective at peak data rate. In this respect, resonant clocking has become an attracting option and widely investigated in low power designs, which utilize some resonant mechanisms for power reduction [7, 15, 18–20, 23, 24, 27].

Resonant clocks can be implemented in different ways such as standing wave [7], rotary wave [15] and LC tank [18, 20, 23, 24, 27]. Standing wave has the limitation of amplitude variation while rotary wave has the limitation of phase variation. Thus, the original CDN needs to be modified accordingly in order to reduce the distortion. On the other hand, LC resonant clocks have the same phase and amplitudes and hence require minimum modification to the CDN.

An illustration of an LC resonant clock is shown in Figure 10.18, which contains a top-level buffer tree connecting to a resonant grid and then the registers. A few inductors are attached to selected nodes in the grid to form LC tanks with the capacitances from adjacent registers. Capacitors can be inserted between the inductors and the ground for DC decoupling and voltage level shifting.

The main idea of LC resonant CDN is to store energy in the distributed LC oscillators and hence reducing the energy dissipation. This concept can be simply explained using Figure 10.19. Consider the circuit shown in Figure 10.19(a). There is no energy stored in the circuit, and the capacitor draws energy from the source every time it needs to charge. Now consider the circuit shown in Figure 10.19(b), where an inductor is attached parallel to the capacitor, forming a parallel LC tank. At resonance, the LC tank oscillates and energy is exchanged between inductor and capacitor. Whenever the capacitor needs to charge, some energy is transferred from

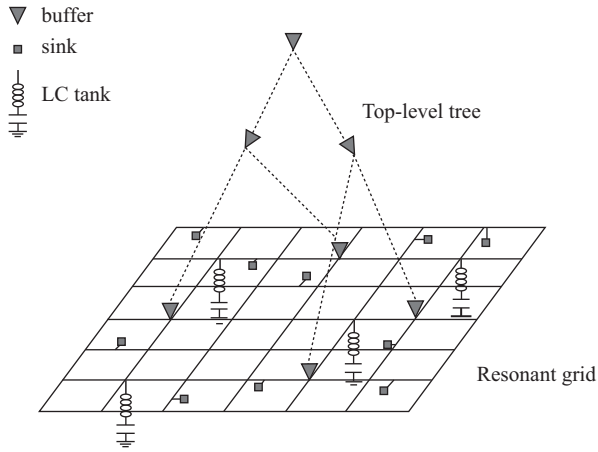


Figure 10.18 Resonant CDN using distributed LC tanks

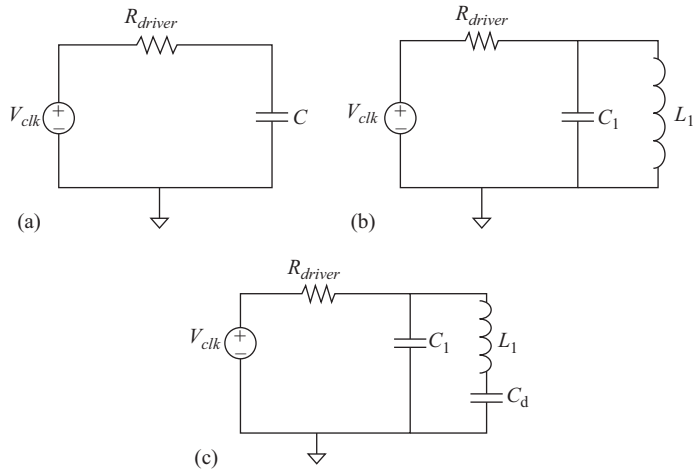


Figure 10.19 (a) RC circuit, (b) LC tank and (c) LC tank with decoupling capacitor (decap)

the inductor, and vice versa. As such, the capacitor draws less power from the source. This exchange of energy between inductor and capacitor occurs at a specific frequency called resonant frequency and is given by

$$f_r = \frac{1}{2\pi\sqrt{L_1 C_1}} \tag{10.1}$$

Ideally, f_r needs to perfectly aligned with the fundamental frequency f_0 of the clock, i.e.,

$$f_r = f_0 = \frac{1}{T_0} \tag{10.2}$$

where T_0 is the period of the clock. The LC tank configuration in Figure 10.19(b) will oscillate between positive and negative amplitudes. For the circuit to operate from 0 to V_{dd} as required by CMOS logic, we must add a positive bias using a decoupling capacitor (decap) in series with the inductor, as shown in Figure 10.19(c). To ensure proper functioning of the circuit, the resonant frequencies of the series LC and the parallel LC should be distinct. Accordingly, careful sizing of the decap is needed [20],

$$\frac{1}{2\pi\sqrt{L_1 C_d}} \ll \frac{1}{2\pi\sqrt{L_1 C_1}} \tag{10.3}$$

In practice, we use $C_d \approx 10C_1$ according to Reference 18. Furthermore, the wires, capacitors and inductors have parasitic resistance, which can damp the resonance. This poses a lower bound on the quality factor of the inductor. However, the biggest challenge to implement resonant clocking technique is the large inductor area overhead.

10.3.2.2 Inductor placement

With the typical clock frequency (2–3 GHz) and the capacitance that needs to be resonant by an inductor ($\sim 20 - -40$ pF) [18], we estimate that a single-turn TSV inductor is sufficient ($\sim$0.1–0.2 nH). Accordingly, we will limit all TSV inductors to single turn, and there is no need to explore the various possible TSV inductor types as in Section 3.2. As an initial exploration, we simply followed the work in Reference 18 to place the regular spiral inductors as desired. We then replaced these inductors with our TSV inductors. Finally, we resized the buffers in order to maintain the full swing of the clock without any excessive buffer sizes. However, the above framework is only good in the ideal case, assuming TSV inductors are available anywhere as desired. In practice, however, this can never happen. Many challenges have to be addressed to apply TSV inductors in the LC CDN such as

- TSV inductors must be configured using idle TSVs, and accordingly their locations, inductance and quality factors are strictly constrained. As such, they may not be optimal or even available as what we desire.
- TSV inductors can be in any orientation with any distance apart, and thus the coupling situation is complicated. These are different from the placement of conventional spiral inductors.

A new framework is needed to opportunistically utilize TSV inductors for maximizing possible power reduction in LC resonant CDNs. This new framework utilization is given in Reference 23.

10.3.2.3 Experimental results

To demonstrate the efficacy of the TSV inductors, we implemented the resonant clocking using an industrial CDN, and the detailed information is reported in Table 10.8. The CDN is designed to work at 3 GHz. After the inductors are inserted, we fine-tuned the buffer sizes so that the max skew is below 100 ps.

Table 10.8 Benchmark information

Design	Sink cap (pF)	Total buffer size (mm)	Power (W)
D1	4.2	172.7	11.4
D2	4.1	103.5	7.1
D3	3.6	192.6	11.6
D4	3.4	73.8	5

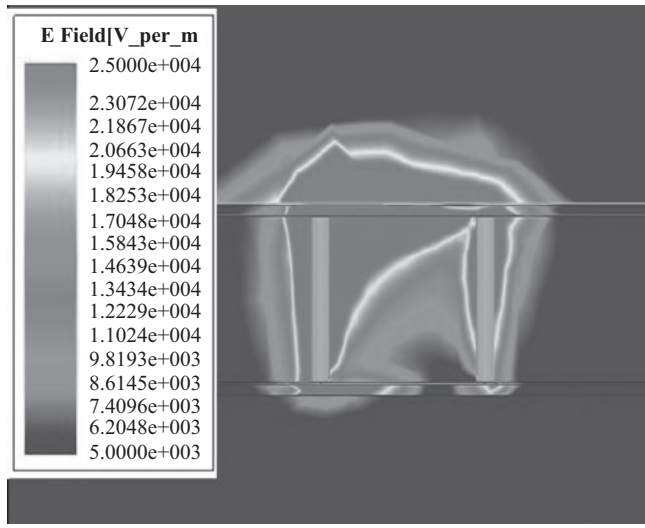
Table 10.9 Footprint/power comparison between TSV inductor based and conventional spiral inductor based resonant CDNs

Design	Total buffer size (mm)	# of inductors	Conventional spiral inductor based CDN		TSV inductor based CDN		
			Footprint (μm^2)	Power (W)	# of idle TSVs	Footprint (μm^2)	Power (W)
D1	126.9	26	988,000 (1)	8.0 (−29.8%)	24,319	166,841 (1/5.92)	7.9 (−30.7%)
D2	62.1	23	897,000 (1)	3.8 (−46.5%)	21,607	147,591 (1/6.06)	3.7 (−47.9%)
D3	120.6	40	1,620,000 (1)	6.7 (−42.2%)	32,591	256,680 (1/6.30)	6.6 (−43.1%)
D4	52.2	12	456,000 (1)	3.4 (−32.0%)	15,925	77,004 (1/5.92)	3.2 (−36.0%)

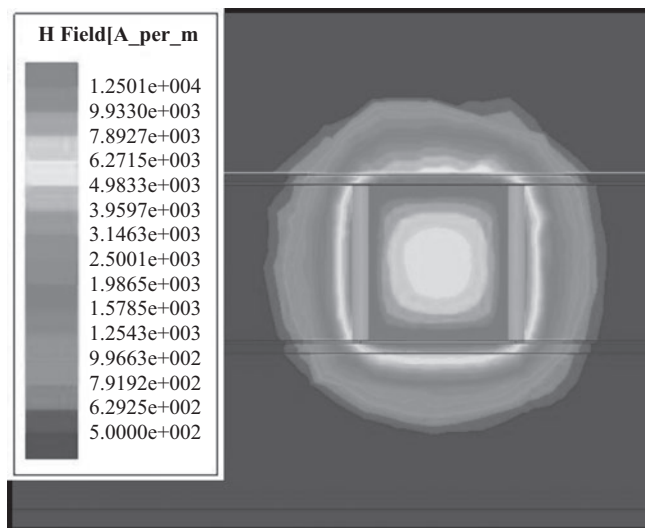
We first compare the inductor area and skew between the TSV inductor based resonant design and the conventional spiral inductor based design [16]. We try to match the power from both methods for fair comparison, and the results are reported in Table 10.9. In the table, we also report the number of spiral inductors used by Yue and Wong [4], as well as the total number of idle TSVs and the number of TSV inductors used. For the power, we also include the reduction percentage over the regular design without inductors. The area is calculated in the same way as described in Section 3.3. From the table, we can see that using TSV inductors can reduce the inductance area by up to 7.7× compared with using the conventional spiral inductors, with the same power reduction. Also, it is interesting to see that only a small number of idle TSVs can be utilized to form TSV inductors. The max skew for the design case is below 100 ps.

10.4 Micro-channel shielding

From Observation 2 in Section III, the substrate loss dominates when the frequency is over 1 GHz and when the substrate conductivity is over 10 S/m (which is normal for digital applications). In other words, the TSV inductor is subject to severe efficiency loss over 1 GHz due to the eddy current in the substrate. To tackle the issue, we are interested in devising effective shield mechanisms to reduce such loss.



(a)



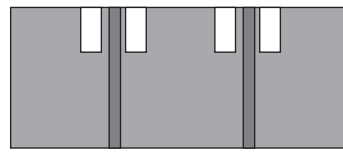
(b)

Figure 10.20 (a) E field and (b) H field without micro-channel distributions

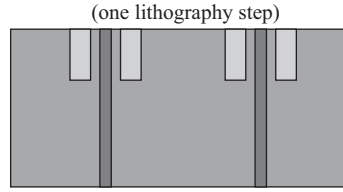
To help understand the distribution of eddy current in the TSV inductor, we simulate it with the nominal setting shown in Section III. The resulting *E* and *H* fields are plotted in Figure 10.20. From the figure, we can see clearly that the *E* field decreases as we get farther from the TSVs, while the *H* field completely penetrates through the area between the TSVs. As such, we can expect that most of the eddy

(a) Chip after FEOL and BEOL processing

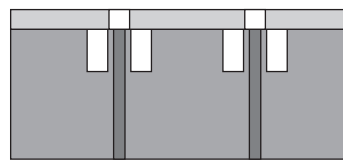
(b) Electrical through via fabrication



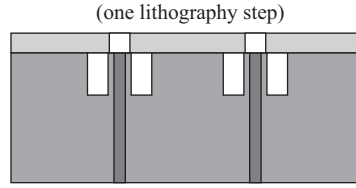
(c) Silicon etch for micro-channels
(one lithography step)



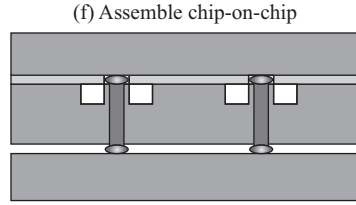
(d) Spin coat and polish a sacrificial
polymer material



(e) Avatrael cover for micro-channels
spin coated, patterned and cured
(one lithography step)



(f) Assemble chip-on-chip



(g) Assemble chip on substrate

Figure 10.21 Micro-channel fabrication steps

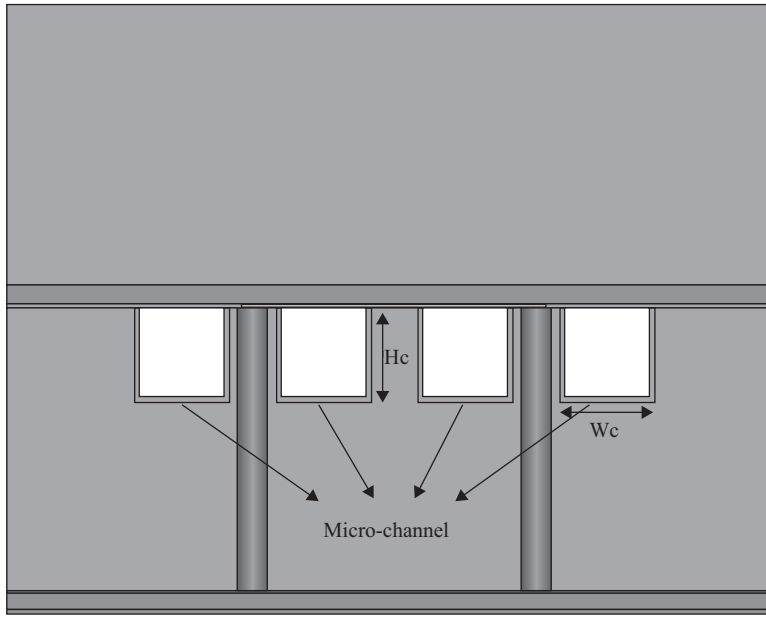
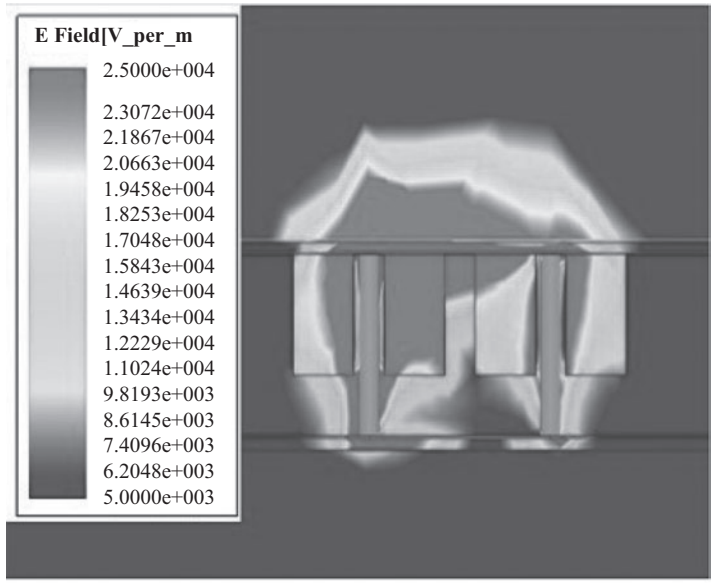


Figure 10.22 Micro-channel shields for substrate loss reduction

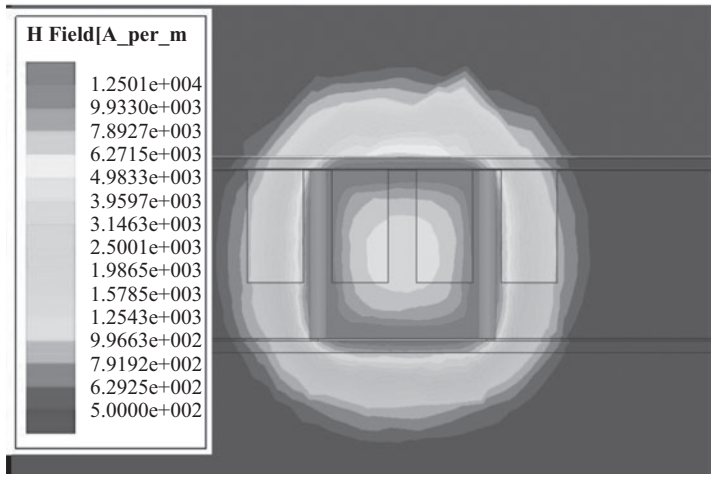
current loss comes from the silicon substrate near the TSVs, which inspires us to think: Why not remove the silicon substrate in that area?

This reminds us about a seemingly irrelevant technique, micro-channel, which has been widely used as a low-cost heat-removal technique in 3D ICs, e.g., References 1, 6, and 31. Simply speaking, the technique etches a channel from the bottom surface of the substrate for liquid cooling and only requires extra two lithography steps, which are relatively cheap to implement. The fabrication process of micro-channel is already mature, an example of such a process from IBM and Nanonexus is shown in Figure 10.21 [6]. In our situation, we can place such channels adjacent to the TSVs to remove part of the substrate. Specifically, we etch four identical channels, one on each side of the two TSVs. An illustration of such a structure is shown in Figure 10.22 for a two-tier design. For multiple tiers, we need to place four channels at each tier, adjacent to the TSVs. The micro-channels can either be filled with coolant-like conventional micro-channels, or just open with air. Note that these channels are etched on the backside of the silicon substrate and will not affect any devices.

The E and H fields for the TSV inductor structure with micro-channels are plotted in Figure 10.23(a) and 10.23(b), respectively. From the figure, we can see that the E field is almost same in the silicon and the E field in the micro-channel does not contribute to losses. Hence the quality factor increases. The H field does not have any impact, and hence the inductance does not vary much for the design with micro-channels compared without the micro-channels case. An extra benefit of such technique is the reduced temperature at the inductor. When inductors are used to form



(a)



(b)

Figure 10.23 (a) E field and (b) H field with micro-channel distributions

antennas, they typically bear high temperature. Accordingly, with the micro-channels the heat will be able to dissipate faster. To verify the effectiveness of this approach, we vary the height (H_c) and the width (W_c) of the four micro-channels and compare the improvement of Q at 10 GHz based on a structure with two tiers ($T = 2$) and six turns ($N = 6$). All the other parameters conform to the nominal settings discussed in Section III. The micro-channels are placed 5 μm from TSV center to the nearer edge

Table 10.10 Q vs. micro-channel dimension (10 GHz, N = 6, T = 2)

		W_c (μm)		
		10	**20**	**25**
H_c (μm)	10	6.71 (4.5%)	6.92 (8.1%)	7.03 (9.6%)
	20	7.02 (9.6%)	7.29 (13.8%)	7.46 (16.3%)
	30	7.34 (14.6%)	7.76 (21.1%)	7.94 (23.9%)
	40	7.73 (20.5%)	8.28 (29.2%)	8.59 (34.0%)
	50	8.25 (28.8%)	8.98 (40.1%)	9.41 (46.1%)
	60	9.12 (42.2%)	10.34 (61.4%)	10.96 (71.0%)

Table 10.11 Q vs. N and T for maximum micro-channel dimensions (at 10 GHz)

Q		***T***			
		2	**3**	**4**	**5**
N	1	10.96 (5.88%)	13.12 (14.5%)	14.53 (38.9%)	14.9 (70.3%)
	2	11.36 (11.7%)	11.31 (78.5%)	7.59 (168%)	6.14 (359%)
	3	11.89 (26.3%)	9.15 (167%)	4.65 (406%)	2 (2034%)
	4	11.89 (42.4%)	7.46 (269%)	2.93 (1007%)	1.03 (N$/%A)
	5	11.37 (55.3%)	5.97 (371%)	1.87 (N$/%A)	0.19 (N$/%A)
	6	10.98 (71%)	4.74 (483%)	1.01 (N$/%A)	−2.08 (N$/%A)

of the channel. The resulting Q is reported in Table 10.10. The improvements over the case without micro-channel shields are also reported in parentheses. From the table, we can easily see that both channel height and width have profound impact on Q. For the maximum height of 60 μm and width of 25 μm, a 71.0% improvement of Q over the TSV inductor without micro-channel can be observed. Considering reliability and manufacturability, the AR of the channel is limited [17], and it depends on the dimensions of the channel and the separation between them. Accordingly, designers should carefully consider the tradeoff between the micro-channel dimension and Q. In the future, we will study the mechanical reliability of the micro-channel shield structure.

We further study how Q and L change when using maximum micro-channel dimensions for different number of turns N, number of tiers T and frequency f. The results on Q and L at 10 GHz are reported in Tables 10.11 and 10.12, respectively. To show the effect at different frequencies, the results on Q at 1 GHz are reported in Table 10.13. Note that we omit the table for L at 1 GHz as it remains constant with or without the micro-channel. In all the tables, improvement over the case of the same N and T but without the micro-channel is also reported in parentheses. From the tables, we can draw the conclusion that micro-channel technique is more important at larger N and T, and at higher frequencies, i.e., both Q and L improve significantly. This is in accordance with the intuition that substrate losses become larger with larger N, T or

Table 10.12 L vs. N and T for maximum micro-channel dimensions (at 10 GHz)

Q		*T*			
		2	**3**	**4**	**5**
N	1	0.135 (0.0%)	0.344 (0.0%)	0.577 (0.0%)	0.828 (1.2%)
	2	0.344 (0.0%)	0.958 (0.0%)	1.729 (0.0%)	2.708 (12%)
	3	0.594 (0.0%)	1.7 (0.0%)	3.523 (40%)	5.882 (1615%)
	4	0.843 (0.0%)	2.741 (1.0%)	5.959 (424%)	8.855 (N$/%A)
	5	1.093 (0.0%)	5.97 (3.39 (2.2%)	8.577 (N$/%A)	3.055 (N$/%A)
	6	1.396 (0.0%)	5.206 (0.0%)	10.634 (N$/%A)	−3.518 (N$/%A)

Table 10.13 Q vs. N and T for maximum micro-channel dimensions (at 1 GHz)

Q		*T*			
		2	**3**	**4**	**5**
N	1	3.03 (0.0%)	3.74 (0.5%)	4.15 (1.0%)	4.44 (2.0%)
	2	3.36 (0.0%)	4.72 (3.2%)	5.37 (4.8%)	5.9 (9.9%)
	3	3.76 (0.2%)	5.28 (1.7%)	6.39 (15.7%)	6.83 (36.4%)
	4	4.02 (0.8%)	6.04 (11.1%)	7.11 (37.4%)	7.57 (88.3%)
	5	4.13 (1.0%)	6.49 (17.8%)	7.65 (67.0%)	7.78 (153.0%)
	6	4.29 (2.7%)	6.81 (26.5%)	7.92 (98.5%)	8.01 (235%)

higher *f*. At 10 GHz, up to 21× increase in Q and 17× increase in L are observed in Tables 10.11 and 10.12 for $N = 3$ and $T = 5$ case, while at 1 GHz only Q is improved by up to 3× is observed when $N = 5$ and $T = 5$ as shown in Table VII.

One more thing worth mentioning here is that the increased SRF brought by the micro-channel shield. For example, in Table 10.11, when $N = 5$ and $T = 4$, the TSV inductor without micro-channels ceases to work as an inductor ($Q < 0$), while the TSV inductor with micro-channels still provides positive quality factor due to the reduced capacitive coupling. For that reason, no improvement is reported for these cases. However, for some cases in Tables 10.11 and 10.12 the reduction of capacitive coupling is not sufficient to make the structure work as an inductor as observed for the $N = 6$ and $T = 5$ case.

Finally, we study how the micro-channel affects the stress distribution. In order to observe the stress distribution with and without micro-channel, we use ANSYS Mechanical software to simulate the stress; we place the micro-channel as close to the TSV as possible to observe the maximum impact. The parameters of the materials used are summarized in Table 10.14. The substrate height is 60 μm, and the micro-channel width (height) is 20 μm (30 μm). The Von Mises stress distributions of the inductor structure with and without the micro-channel are shown in Figure 10.24.

Table 10.14 Parameters and respected values used in the stress simulation

Parameter	Copper	Silicon dioxide	Silicon
Elastic modulus (GPa)	110	71	160
Coefficient of thermal expansion (μK^{-1})	17	5	2.3
Poisson ratio	0.35	0.16	0.28

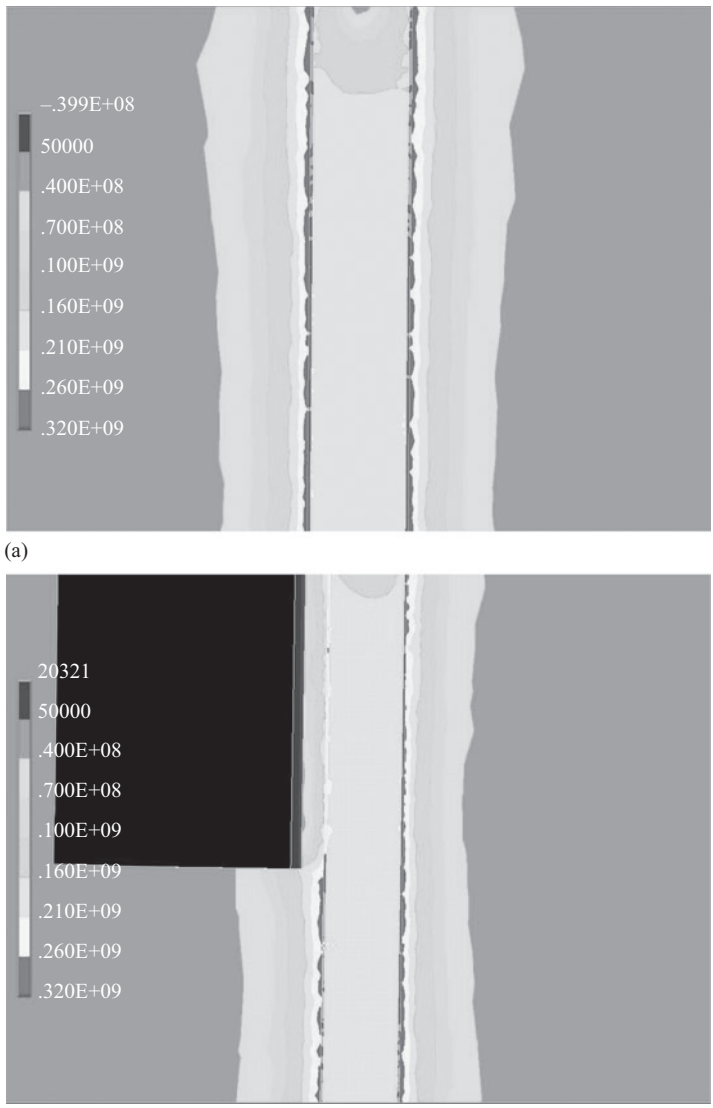


Figure 10.24 Stress distribution (a) without micro-channel and (b) with micro-channel

From the figure, we can observe that the stress is actually decreased, which is mainly due to the expansion allowed along the channel surface. Therefore, the micro-channel has the extra benefit of stress reduction.

10.5 Summary and conclusions

In this chapter, we systematically examined how various parameters affect their performance. We then studied the possible low-frequency applications of TSV inductors to inductive DC–DC converters and resonant clocking. For the former, experimental results show that by replacing conventional spiral inductors with TSV inductors, with almost the same efficiency and output voltage, up to 4.3× inductor area reduction can be achieved. For the latter, our scheme with TSV inductors can reduce the inductor area by up to 7.7× with the same power consumption. In addition, we proposed a novel shield mechanism utilizing the micro-channel technique to drastically improve the quality factor and the inductance at higher frequencies.

References

[1] B. Shi and A. Srivastava, "TSV-constrained micro-channel infrastructure design for cooling stacked 3D-ICs", *ACM International Symposium on Physical Design*, pp. 113–118, 2012.

[2] B. Zhang *et al.*, "3D TSV transformer design for DC–DC/ACDC converter", *60th Electronic Components and Technology Conference (ECTC)*, pp. 1653–1656, Jun. 2010.

[3] B. Zhai *et al.*, "Theoretical and practical limits on dynamic voltage scaling", *Design Automation Conference*, 2004.

[4] C. P. Yue and S. S. Wong, "On-chip spiral inductors with patterned ground shields for Si-based RF ICs?", *IEEE Journal of Solid-State Circuits*, vol. 33, no. 5, pp. 743–752, May 1998.

[5] D. H. Kim *et al.*, "3D-MAPS: 3D massively parallel processor with stacked memory, Solid-State Circuits Conference Digest of Technical Papers (ISSCC)", *IEEE International*, pp. 188–190, 19–23 Feb. 2012.

[6] D. Sekar *et al.*, "A 3D IC technology with integrated microchannel cooling", *Interconnect Technology Conference, IITC*, 2008.

[7] F. O. Mahony, C. P. Yue, M. A. Horowitz and S. S. Wong, "A 10-GHz global clock distribution using coupled standing-wave oscillators", *Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 1813–1820, 2003.

[8] G. VanAckern, "Design guide for CMOS process on-chip 3D inductor using thru-wafer vias", Master Thesis, 2011.

[9] H. Fujiwara *et al.*, "A two-port SRAM for real-time video processor saving 53% of bitline power with majority logic and data-bit reordering", *Proceedings of the 2006 International Symposium on Low Power Electronics and Design, 2006. ISLPED'06*, pp. 61, 66, 4–6 Oct. 2006.

[10] Huang *et al.*, "Interleaved three-dimensional on-chip differential inductors and transformers", US Patent 2008/0272875, Nov. 2008.

[11] P. D. Franzon, W. R. Davis and T. Thorolffson, "Creating 3D specific systems: architecture, design and CAD", *Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1684–1688, 8–12 March 2010.

[12] I. Loi, F. Angiolini, S. Fujita, S. Mitra and L. Benini, "Characterization and implementation of fault-tolerant vertical links for 3-D networks-on-chip", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 1, pp. 124–134, Jan. 2011.

[13] J. Oh and M. Pedram, Gated clock routing minimizing the switched capacitance, Design, Automation and Test in Europe, 1998, pp. 692–697.

[14] J. Wibben *et al.*, "A high-efficiency DC–DC converter using 2 nH integrated inductors", *Proceedings of the IEEE Journal of Solid-state Circuits*, vol. 43, no. 4, pp. 844–854, 2008.

[15] J. Wood, T. C. Edwards and S. Lipa, "Rotary traveling-wave oscillator arrays: a new clock technology", *Journal of Solid-State Circuits*, vol. 36, no. 11, pp. 1654–1664, 2001.

[16] J. Zhang, "Inductor with patterned ground plane", US Patent 2009/0250262, 2008.

[17] K. Gantz and M. Agah, "Predictable three-dimensional microfluidic channel fabrication in a single-mask process", *Technical Digest of the 14th International Conference on Solid-State Sensors, Actuators, and Microsystems (Transducers07)*, pp. 755–758, Jun. 10–14, 2007.

[18] M. R. Guthaus, "Distributed LC resonant clock tree synthesis", *International Symposium on Circuits and Systems (ISCAS)*, pp. 1215–1218, 2011.

[19] R. Somayyeh *et al.*, "Design of resonant clock distribution networks for 3-D integrated circuits", *Proceeding of: Integrated Circuit and System Design. Power and Timing Modeling, Optimization, and Simulation – 21st International Workshop, PATMOS* 2011.

[20] S. Chan, P. Restle, K. Shepard, N. James and R. Franch, "A 4.6GHz resonant global clock distribution network", *International Solid State Circuits Conference*, pp. 342–343, Feb. 2004.

[21] U. R. Tida, C. Zhuo and Y. Shi, "Novel through-silicon-via inductor based on-chip DC–DC converter designs in 3D ICs", *ACM Journal of Emerging Technologies*, Vol. 11(2), no. 16, November 2014.

[22] U. R. Tida, C. Zhuo and Y. Shi, "Through-silicon-via inductor: is it real or just a fantasy", *Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific*, vol., no., pp. 837, 842, 20–23 Jan. 2014.

[23] U. R. Tida, V. Mittapalli, C. Zhuo and Y. Shi, "Opportunistic through-silicon-via inductor utilization in LC resonant clocks: concept and algorithms", *IEEE Computer Society Annual Symposium on VLSI, ISVLSI 2014*, Tampa, FL, USA, July 9–11, 2014.

[24] U. R. Tida, V. Mittapalli, C. Zhuo and Y. Shi, '"Green" on-chip inductors in three-dimensional integrated circuits', *The IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2014*, San Jose, CA, USA, November 3–6, 2014.

[25] U. R. Tida, C. Zhuo, R. Yang and Y. Shi, "On the efficacy of through-silicon-via inductors", *in IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 7, pp. 1322–1334, July 2015.

[26] X. Bian *et al*., "Simulation and modeling of wafer level silicon-base spiral inductor", *Electronic Packaging Technology and High Density Packaging (ICEPT-HDP)*, 13th International Conference, 13–16 Aug. 2012.

[27] X. Hu and M. Guthaus, "Distributed LC resonant clock grid synthesis", *IEEE Transactions on Circuits and Systems I*, vol. 59, no. 11, pp. 2749–2760, Nov. 2012.

[28] X. Zhao *et al.*, "Analysis of DC current crowding in through-silicon-vias and its impact on power integrity in 3D ICs", *Proceedings of Design Automation Conference (DAC) 49th ACM/EDAC/IEEE*, pp. 157, 162, 2012.

[29] Y. Chio *et al.*, "DC–DC converter-aware power management of low-power embedded systems", *Proceedings of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 8, pp. 1367–1381, 2007.

[30] Y. I. Bontzios, M. G. Dimopoulos and A. A. Hatzopoulos, "Prospects of 3D inductors on through silicon vias processes for 3D ICs, VLSI and System-on-Chip (VLSI-SoC)", *IEEE/IFIP 19th International Conference*, pp. 90–93, 3–5 Oct. 2011 .

[31] Y. J. Kim *et al.*, "3D-MAPS: 3D massively parallel processor with stacked memory", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, IEEE International, pp. 188–190, 19–23 Feb. 2012.

[32] Z. Tao *et al.*, "A 3D SoC design for H.264 application with on-chip DRAM stacking", *3D Systems Integration Conference (3DIC)*, 2010 IEEE International, 2010.

Chapter 11

3D NoC: a promising alternative for tomorrow's nanosystem design

Prasun Ghosal[1], *Tuhin Subhra Das*[1], *Soumyajit Poddar*[1], *Munshi Mostafijur Rahaman*[1] *and Avik Bose*[1]

Four primary aspects of chip design are processor, memory, IO, and communication. Communication amalgamated over a SoC (system-on-chip) is the basis of origin of an NoC (network-on-chip). Continuous increase in processing/communication needs with the rapid growth in VLSI industry providing higher and higher integration density within a single die has boosted the step towards this new paradigm shift by the researchers. Advent of three-dimensional (3D) integrated circuits (ICs) design technologies has given this direction another positive thrust. Researchers are indeed very hopeful with the amalgamation of these two different technologies and thereby in developing new methodologies for designing 3D NoCs to cater the need of high-performance nanoscale computing and communication systems tomorrow. This chapter describes different design challenges, available technologies, design and performance issues and parametric measurement of such nanoscale systems, emerging cutting-edge technologies, and possible future directions in designing 3D NoC-based nanosystems.

11.1 Introduction

Since mid-1960s advancements in IC technologies have led researchers to incline their thinking towards boosting the performance of microprocessors exploiting high transistor density. With the advent of VLSI, at that higher integration level, it was necessary to employ parallelism to extract the efficiency out of the processing engine as much as possible. Several techniques had been employed; most popular ones are instruction level parallelism, dynamic branch prediction, out of order execution, speculative execution, and thread or more coarsely process level parallelism. Later two approaches needed complex compilers and operating system supports, whereas complex processor hardware designs were inevitable for the formers. Past few decades

[1]Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, West Bengal, India

witnessed the evolution of dominant technologies like SMT (simultaneous multi-threading) [1], CMP (chip multi-processor system) [2], and MPSoC (multi-processor system on chip) [3]. Long interconnect delays forced the micro architectural design of processors to be partitioned into small localized processing elements (PEs). For this reason, CMP was the obvious choice as it was already decomposed into individual processing cores. This fueled the shifting of focus of research communities to a completely new paradigm other than processing; that is, communication, as it became an indispensable aspect of multi-processor architectures. Initially bus was selected for communicating medium. But traditional bus-based interconnection was facing major problems like limited bandwidth, critical arbitration, timing synchronisation issues, energy inefficiency, and long wire delays [4]. In the early 21st century, NoC came into picture as a promising solution for designing large-scale multi-core system into smaller footprint. It supports GALS (globally asynchronous locally synchronous) architecture that facilitates designer to make system more scalable by connecting two different clock domains, which are mesochronous or completely asynchronous to each other in nature.

11.1.1 NoC basics

Achieving increasingly optimized critical paths has made processor clock rates to rise exponentially. Resources on a processor chip those communicate among each other in a single cycle must be physically close. Additional pipelined stages are required to suppress long interconnect delays, especially, when it falls on a critical path [2]. Bus-based interconnect system was facing many challenges to support such communication centric architectural designs: most importantly poor communication latency and power consumption. Heterogeneous MPSoCs with variety of dedicated interfaces have made designs so complex that achieving the separation between computation and communication was too difficult using buses.

Concept of NoC has been developed to mitigate these limitations stated above. By facilitating on-chip communication with higher communication bandwidth, concurrent execution of PEs, incurring modularity in system designing, and effective special reuse of resources NoC provides a breakthrough. Figure 11.1 depicts a two-dimensional (2D) mesh-based NoC topology. PEs may include processor, co-processor, accelerator, application-specific Intellectual Property (IP), peripheral controller, memory, and DSP blocks, etc. and are connected via on-chip interconnections through routers. Here, network interface (NI) units (see Figure 11.2) act as intermediary system between the routers and PEs and are responsible for generating, transmitting, and receiving of data packets among IP cores.

11.1.2 Transition towards 3D

Overall chip area has become a major issue in 2D NoC design. Increasing number of processing cores requires inclusion of more cache memory blocks that in turn requires larger floor design. This has a significant impact on the development as in most systems overall chip area plays an important role. Systems with no such stringent

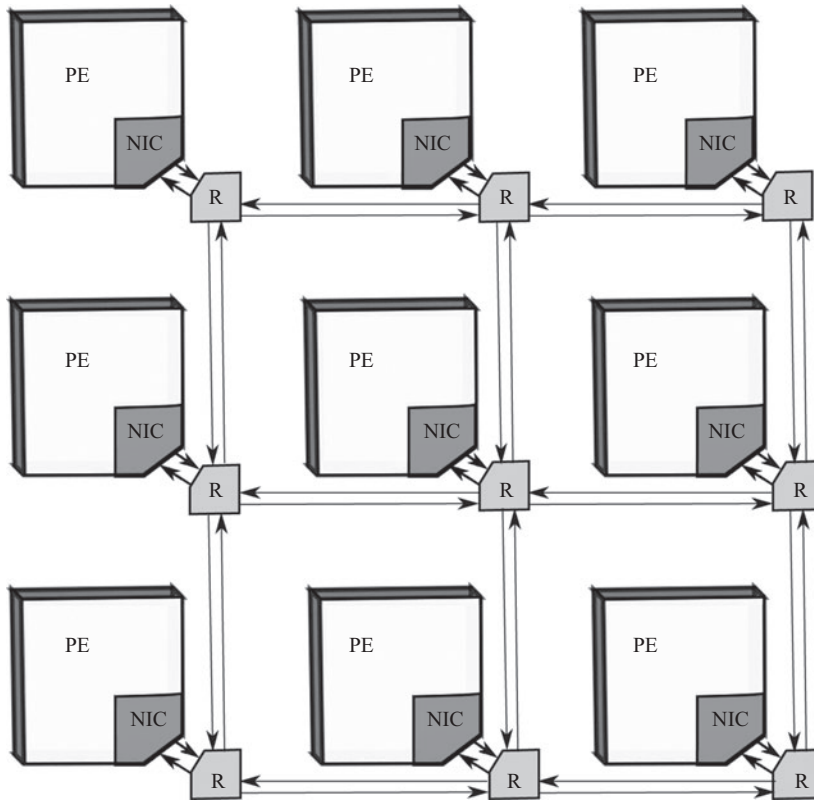


Figure 11.1 Tile-based 2D mesh topology

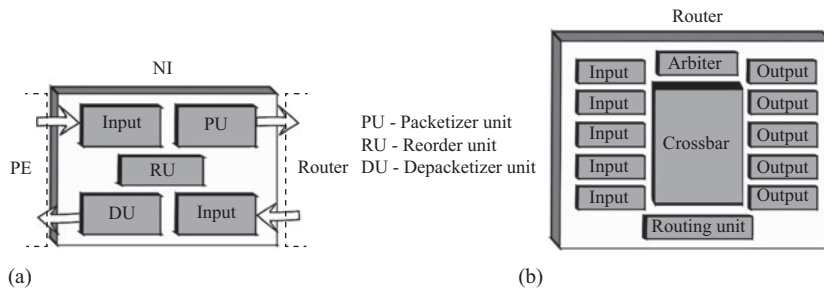


Figure 11.2 Generic NoC. (a) NI and (b) router architecture

area constraint result in increased length of interconnects that strikes over the basic concept of NoC. It affects the overall network latency. 3D NoC offers higher scalability within a limited chip area. As opposed to increase the floor area horizontally with a single layer of circuitry, there are a number of layers placed vertically maintaining

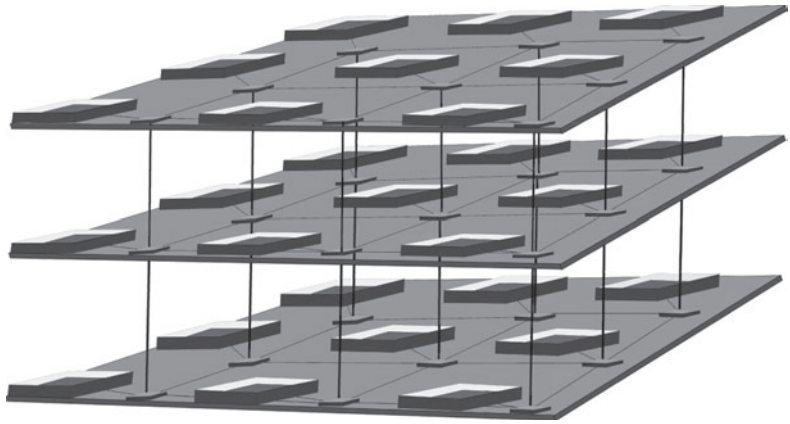


Figure 11.3 3D mesh-based NoC

the chip area restrictions in 3D NoC design. Figure 11.3 shows a mesh-based 3D NoC architecture possessing three layers. Inter-layer routers are vertically connected. The length of a vertical interconnect is much more lower than the horizontal ones connecting two far placed IP blocks. With comparatively lower chip area, intra-layer latency is also improved. The subsequent sections discuss challenges that are being faced and progresses upon them till now in 3D NoC development.

11.2 Design challenges in 3D NoC

The holistic research problems in this domain can broadly be categorized into four different classes.

- First, the communication infrastructure; such as, network topology, router architecture, buffer optimization, link design, clocking, floor planning, and layout.
- Second, the communication paradigm including routing policies, switching techniques, congestion control, power and thermal management, fault tolerance, reliability, etc.
- The third dimension involves designing an evaluation framework for NoC to have a good understanding of achievable throughput, latency, and bandwidth of the network.
- Finally, once the communication infrastructure and paradigm for a NoC have been finalized, a major challenge in overall system design is to associate the IP cores implementing tasks of an application with the routers.

Following subsections depict the above challenges at a glance, by classifying them into two parts, viz. design challenges and technology challenges.

11.2.1 Design challenges

Challenges in 3D NoC design are almost similar to conventional 2D NoC. However extra care has to be taken because of additional wafer layers those are stacked over chip surface. Details of conventional 2D NoC design challenges are omitted due to paucity of space as it is out of scope. Basic design challenges in 3D NoC are as follows.

11.2.2 Macro-architecture

Topology

Topology defines the basic of communication infrastructure among the tiles (IP cores) connecting them directly or through routers within the chip surface. Orientation of this connection may be regular or irregular type. In 3D NoC, connections of different stacked layers are implemented using shorter vertical interconnection access (via) that offers smaller footprint compared to 2D NoC architecture. Here, network diameter, link complexity, node degree, and bisection width are considered as most important topology centric design parameters.

Routing

Routing defines basic logic of forwarding data packets among the communicating cores across the entire chip network. Routing logic must be as simple as possible to minimize the computation as well as communication latency. Deadlock, live-lock, and starvation [5] are some unintended situations that need to be considered while proposing routing methodology. Channel contention and congestion are other two important factors influencing performance and are mostly affected by routing.

11.2.2.1 Micro-architecture

In NoC, micro-architectural optimization issues include design of two important components, viz. the router and NI controller (NIC). With the notion of 3D NoC designing vertical interconnects is another important aspect that needs special care. The task of designing a router encompasses majority on three things. Firstly, the input buffers where the flits (unit of data packet) are stored before routing. Secondly, the arbiter that picks up the flits from the buffer one by one according to a specified logic. Lastly, the crossbar [6] that places flits to proper output channels for the next hop routing, in order to reach the destination. Designing NIC is also investigated in terms of their architectural structure. Suitable addressing schemes are needed for this NIC as to route packets in vertical direction along with horizontal direction specifically for 3D NoC router.

Power consumed by an unit is another important factor that heavily depends upon required cycles to drive an unit for a single operation. Efficient power modelling for each component is required to optimize power requirements. Various performance measurement parameters like routing computation delays, crossbar delays, over all communication latency, throughput, flit injection, and acceptance rates verify the performance of designed router and NIC. Reliability issues encircle the idea on

ability of communication infrastructure to handle errors during transmitting data over the network. Here, variability refers to the discrepancy between expected and actually observed data. It results from the manufacturing imperfections, called process variation. With shrinking feature size of die, there lies a possibility of having a significant gap between the expected and manufactured characteristics. Careful micro-architectural design is needed to address these issues especially for 3D NoC design.

11.2.2.2 Performance

Performance of an NoC is determined in terms of packet latency, network throughput, thermal responsiveness, and energy efficiency. These are defined as follows.

Throughput

Average network throughput quantifies the amount of data received by the receiver node per cycle time. Maximum throughput is obtained when every communicating channel operates with its maximum bandwidth capacity. Besides available bandwidth throughput depends heavily on routing logic and available flow control policy.

Latency

Packet latency is measured by total traversal time consumed by packet to reach destination node from originating one. On reaching its destination a packet has to go through some switches and other interconnect medium, where some computation delay are added along with the transport delay. However, packet delay for same pair of source and sink may vary depending upon various parameters like routing logic, congestion status of the network, etc. 3D NoC architectures have low network diameter compared to their 2D counterpart and offer lower packet latency.

Thermal awareness

This implies how seamlessly a circuit operates on a dense 3D ICs at higher frequency. It has been observed that power density per unit area increases dramatically when dies (2D) are stacked vertically one upon another. With this increasing power density, the 3D chip becomes more heated, increasing the chance of breaking it down at some threshold temperature value. Several thermal-aware and dynamic thermal management techniques [7], [8] have been proposed recently to improve performance offered by 3D NoCs.

11.2.2.3 Physical design

To offer better SoC platform for dense chip integration, NoC makes complex interconnection structure easily manageable by reducing wiring complexity level. Here, floor planning, placement, and router architecture optimization are solely considered as important physical design metrics. With the increasing network dimension inclusion of 3D NoC design has become almost inevitable for designing large-scale interconnection networks. Here, router plays a crucial role by offering vertical communication between different computational units. The vertical interconnection technologies usually refer to the use of micro-bumps, wire bonding, wireless interconnects using capacitive or inductive coupling, and through-silicon vias (TSVs). Out of which

TSVs offer very-high-density vertical interconnects and supposed to be most promising technology [9], [10] above all. This allows NoC design in smaller area footprint. But the TSV fabrication technologies are not mature enough due to the TSVs intrinsic size and the pitches that are necessary to ensure the correct functionality of the required area around them [9]. And due to the low fabrication yield of TSVs [10], the number of fabricated TSVs in each chip is limited as it increases the cost in terms of area and manufacturing cost. So there should be a limit on the number of vertical data links to avail the benefit of 3D NoC architecture.

11.2.2.4 Application mapping

Mapping application is the first and one of the most critical parameters in 3D NoC design flow, firmly affecting the 3D NoC performance and cost. The mapping of cores onto NoC architecture makes new challenges. Sepulveda *et al.* [11] show significant enhancement on power and latency of the 3D NoC compared to previous. At the time of 3D NoC design, optimal mapping of on-chip components is very essential. For this researchers proposed mapping algorithms to design NoC, intending various performance issues. More temperature increases in 3D NoC rather than in traditional designs. Due to mapping of multiple tasks on a single PE after partitioning increases the computation power. Addo-Quaye [12] explored thermal-aware mapping algorithm for 3D NoC to reduce communication and peak temperature. Energy savings can be handled effectively by appropriately assigning functionalities of application to layers. Siozios *et al.* [13] have shown this by taking high-level mapping strategy for 3D NoC architectures with different supply voltages and followed optimized bandwidth-constraint approach which provides communication reduction. In order to distribute traffic evenly on the links for a particular application, the bandwidth constraints in the NoC architecture need to be satisfied by the mapping process.

The mapping of the switches onto the layers is considerable important constraint for many 3D fabrication technologies. Mapping techniques also help to gain optimality in the number of TSVs and power consumption in 3D NoC design. One of the objectives of Task Mapping is to utilize the execution parallelism in the target task set, and data reusing among task, deciding which tasks should be mapped to the same PE. Determine the association of routers of the fabric to the cores of an application is a significant challenge in 3D NoC-based system design. Tenable improvement in communication cost, dynamic performance, and energy consumption has produced by using well-suited mapping strategies for 3D mesh-based NoC in Reference 14.

11.2.2.5 Energy-aware modelling and design

Along with thermal issue, energy consumption is another challenge, especially for battery-driven systems. Energy is a major considerable issue when allocating thermal-constrained tasks on 3D NoC. Energy saving design covers a large portion in 3D NoC design process. Decreasing power consumption by adopting thermal-aware router architecture results in significant transfer energy reduction. A thermal-aware, energy saving router sharing architecture for 3D NoC has been proposed in Reference 15. Formulation of energy-efficient tasks mapping strategy in 3D NoC interconnection

communication structure aids to save interconnect energy. As a consequence, it has a great impact on the maintaining system performance and reliability. Significant improvements can be achieved in energy consumption, adopting efficient techniques in circuit design style of 3D NoC as proposed in References 16 and 17.

11.2.2.6 Thermal issues

3D die stacking with TSV brings notable improvement over 2D NoC by minimizing the length of the global long interconnection wire and mitigating power consumption on long wires and also reducing signal delay. The common problem in 3D stack is the heat dissipation of inner active layer because power density per unit area increases dramatically when 2D dies are stacked one upon another. Heat generated by inter-active layer might cause a large temperature gradient that leads to hotspot region within the chip area. To balance temperature distribution, various methods of floor-plan optimization and wire routing are proposed in References 18–20.

Traffic distribution also has an important role over controlling temperature variation over chip surface. Because balanced traffic distributed on the network does not guarantee that the temperature distribution over chip is also even. A traffic-thermal mutual coupling co-simulation platform is required for that as described in Reference 21.

Router micro-architecture and routing algorithm should be well designed to achieve the temperature requirement [22], [23]. Besides this thermal-aware hardware design, temperature management with software assistance especially OS-level task scheduling is becoming hot for multi-core processors based system in recent years [24], [25].

11.2.2.7 Reliability analysis

As the technology scales into the deep subatomic regime, NoCs become increasingly more prone to various noise sources. Errors in on-chip communication are induced by crosstalk, coupling noise, spurious voltage, and electromagnetic (EM) interference. Various soft as well as hard faults or error may upset the network generated from router and inter-router links. Channel disturbances or link errors are mostly caused by coupling noise or crosstalk and managed by Error Correction Code (ECC) scheme (like Forward Error Correction [26], Hsiao SEC-DED Coding [27]) followed by Crosstalk Avoidance Codes (CAC) and some compensate technique like retransmission of failure or corrupted message data in the network. In addition to soft faults on-chip interconnects also require proper strategies to deal with hard faults in various module. Deadlock free fault tolerant routing algorithms and redundant hardware help to tolerate this kind of faults.

11.2.2.8 Fault tolerance

A fault may appear in the network in both manufacturing and post-manufacturing processes [28] due to the failure of links or failure of buffer or switch allocator or because of the failure of routing computation. However, failure may be permanent

static faults or dynamic faults. Permanent static faults such as open-circuit and short-circuit faults may appear in the network because of the uncontrolled variations during manufacturing process. Whereas dynamic permanent faults appear at any instant of time because of electromigration and stochastic ageing effect and cannot be discovered in the post-fabrication testing. However, proposed routing should be adaptive in order to tolerate this faulty component and to keep system running on such unavoidable situation.

11.2.3 Emerging technological challenges

11.2.3.1 Wireless NoCs

In order to develop scaling by 3D integration technique, several innovative solutions on inter-chip and intra-chip interconnections have been evolved. Traditional wire-based connection has some limitation like large footprint, long wire delay, and high fabrication cost. To overcome these issues, wireless interconnects using capacitive coupling of small pads [29] or inductive coupling of on-chip spiral inductors [30] are introduced in chip architecture. With the former techniques, expected requirements of over 1000 connections between chips cannot be realized in practice due to large power consumption. Global wireless interconnection utilizes high-frequency EM wave transmission using integrated antennas [31]. Another bottleneck for 3D integration lies in the processing algorithm and architecture of conventional Neumann computers. Wireless on-chip interconnects are a radio-frequency (RF) alternative to metal interconnects for global communication on an IC. RF interconnect channels are based on the followings.

- On-chip micro-strip transmission lines [32]
- On-chip antennas [33]
- On-chip inductors based inductive coupling [34]
- On-chip capacitors based capacitive coupling [35].

Wireless interconnects are implemented in conjunction to provide hybrid communication structures and NoCs, particularly for 2D or 3D MPSoCs. The design of the wireless RF interconnects for multi-core systems in particular requires considerations across a vast variety of subjects including, EM theory, network theory, wireless communication, VLSI design and design automation.

In general, the transmission of RFs is allowed for broadcast and multi-cast communications. In traditional metallic interconnection, small distances (millimetre range) on-chip communication require less transmitted power compared to long distance (metre range) communication. While wireless interconnect provides high band width, low-latency-based energy saving communication for global (long distance) interconnects [36]. Wireless interconnects provide some unique benefits, such as

- cost effective CMOS compatible communication architecture
- reducing area overhead with no wires or waveguides, and
- edge to edge transmission across the chip with reduced consumed energy.

11.2.3.2 CNT, graphene, and other emerging technologies

CNT and graphene-enabled wireless NoC (GWNoC) communications establish the Terahertz band as the frequency band for the operation of future nanodevices (0.1–10 THz). Novel nanomaterials such as graphene are enabling the development of miniaturized EM transceivers [37]. Graphene-based RF nanocomponents [38] and graphene-based nanoantennas [39] set the terahertz band (0.1–10.0 THz) as the expected frequency range of operation for the future EM nanotransceivers. The terahertz band provides a huge bandwidth in the short range that can support very high transmission rates, up to hundreds of terabits per second.

It provides core-level communication due to reduced size of graphene antennas, enabling the integration of one or more of them within an individual computing core. As the information transmitted wirelessly can be potentially received by any receiver within the transmission range, GWNoC natively implements broadcast and multi-cast. Such approach also allows simultaneous transmissions and the creation of reconfigurable communication schemes.

In the RF approach, according to the classical antenna theory, the reduction of the antenna size down to a few hundreds of nanometres would impose the use of drastically high resonant frequencies. Graphene-based nanoantennas are proposed to allow nanosystems to transmit and receive information using graphene-enabled wireless communications (GWC) system. GWC may represent a breakthrough in the research areas of wireless on-chip communications.

GWC offers not only the transmission of information at extremely high speeds but also the design of ultra-low-power and low-complexity schemes. Size of graphene antennas can be greatly reduced compared to metallic antennas with the same resonant frequency which provides the integration of antennas within individual processing cores and the implementation of core-level wireless communication.

This approach can resolve the latency and power bottlenecks of traditional on-chip communication. As a result, the cost of operations such as data coherency, consistency, or synchronization that represent the primary limiting factors in devising nanoscale multi-core architecture could be significantly reduced and, in a few cases, eliminated.

11.2.3.3 Photonic NoCs

Photonic ICs have a wide spectrum of applications like high-performance computing, telecommunications, healthcare and defence [40]. In the last decade, researchers have developed and fabricated feasible solutions for on-chip optical interconnects [41]. Some of the devices developed for photonic IC are also applicable to the VLSI domain (e.g. multi-core and manycore ICs). Recently, high-performance embedded systems based on photonic interconnect are in the research phase [42]. Challenges in Photonic interconnects and thereby photonic NoCs (PNoC) are described in details in subsequent sections.

11.3 Performance centric design of 3D NoCs

Function of NoC is to deliver a massively parallel on-chip communication scenario for today's high-performance portable computing devices. Here, 3D NoC has come up with smaller delay, smaller logic area, less number of repeaters and to meet higher throughput requirements. Important performance centric parameters that have to be considered when devising an NoC architecture for specific application are as follows.

11.3.1 Interconnection topology development

Numerous types of 2D-based NoC topologies are available in the web, though few of them have been implemented in real chip manufacturing process, and some of them have been selected and proposed for extending architecture into 3D NoC. Here, major challenge lies on providing interconnection among the component across different silicon layers and within the same layer in an efficient way. Connection between stacked chip may be wire-bonding or photonic or wireless.

3D NoC topologies can be broadly classified into grid-based topology and tree-based topology. Though most of them follow a regular pattern, however they may be irregular type also. Network connection may be direct or indirect type. In case of direct network, nodes (or tiles) which consist of processing core are directly connected to the network. On the contrary in case of indirect network, nodes consisting of a processing core and an NI are indirectly connected to the network via routers.

Regular grid-based topology

Mesh [43] is commonly known grid-based 2D regular topology which has been extended easily in 3D micro-architecture domain. And TSV is a common choice which cuts across thinned silicon substrates to establish inter-die connectivity. A vertically partially connected 3D mesh [44] architecture proposed in Reference 45, which targets topologies with irregularly placed vertical connections in a deadlock free manner.

Tree-based topology

A 3D layout method that divides the planar network into several parts and connects them by using vertical links in order to reduce their wire length. 3D layouts of Fat Tree [46], Fat HTree [46], and low latency-based scalable BFT have been proposed in Reference 47, where original 2D layout of a given Fat Tree is divided into multiple tiers and then these tiers are stacked together using vertical interconnects.

11.3.2 Routing policy

Routing algorithm determines the message forwarding path between two communicating nodes in the network. Routing is normally classified into source and distributed routing. In source routing, the routing path is decided at the source node prior to the injection of packet in the network. Here, routing path information is carried with

the data packet and saves the routing computation time. But at the same time, this increases bandwidth requirement due to the increase in payload size. In distributed routing, the path is computed at each intermediate node. Distributed routing is also classified into oblivious and adaptive routing.

Oblivious routing

Routing path is completely determined by source and destination address in case of Dimension Order routing or Oblivious routing. For 3D cuboid, the routing is made deadlock free by restricting six turns out of total 24 possible 90 degree turns. West South First [48], North Up Last [48], Negative First [48] routings are example of commonly used Oblivious routing. Dimension Order routing like XYZ and ZXY [49] routings are widely used due to simplicity in router implementation and yielding low latency. Dimension Order routing routes packet to one dimension at a time. However, its throughput decreases rapidly with increasing load even for few hops. PROM3D [50] gives diverse path with almost no hardware cost overhead. Aim is to design a routing algorithm for 3D NoC which is an oblivious and minimal. Path decisions are taken locally and distributed randomly among all minimal paths. Variants of PROM (like Uniform, Parametrized, Variable parameterized) are available depending on random path distribution over the network.

Adaptive routing

Path diversity or degree of adaptiveness is defined by number of possible routing paths between source and sink node. Adaptive routing is congestion or fault aware or thermal aware and performs better under non-uniform and highly loaded traffic. Though fully adaptive routing does not guarantee the deadlock freeness, so detection of deadlock and its recovery becomes an important issue here. In addition to that, adaptive routing does not guarantee in-order packet delivery at destination end and also increases switching complexity, as arbiter has to choose a single channel from a set of alternative channel. Chao *et al.* [49] have proposed a thermal-aware adaptive routing to ensure thermal safety for throttled 3D NoC using a proactive downward routing. Another dynamic programming-based runtime thermal management (DPRTM) policy has been proposed by Al-Dujaily *et al.* [52] to design a runtime thermal regulation strategy. Thus runtime thermal management strategy effectively diffuse and manage heat of 3D chip for a better throughput performance in networks on chip (NoC). But the work is limited to symmetric 3D NoC. Another group has also worked with 3D hybridization architecture. Rahmani *et al.* [53] have proposed hybridization policy to mitigate vertical bottleneck. AdaptiveZ, a congestion-aware inter-layer routing algorithm, has also been proposed by them which selects a vertical channel adaptively according to traffic condition and selects XY routing when packet lies in the target layer. Rahmani *et al.* [44] have proposed an AdaptiveXYZ routing, a updated version AdaptiveZ routing for monitoring traffic, thermal management, and fault tolerance. While Chaochao *et al.* have proposed a low overhead fault-aware deflection routing [54] to deal with faulty links of a 3D stacked mesh NoC.

11.3.3 Flow control mechanism

Flow control policy in NoC routing determines the process of storing and transferring data packets, passing through the router over the network. The network is usually termed as either circuit switch or packet switch based network based on the chosen switching policy. A circuit switched network reserves a physical path before transmitting the data packets, while packet switched networks transmit the packets without reserving the entire path. Packet switched networks can further be classified as Store and Forwarding, Virtual Cut Through, and Wormhole switching (described in details in Reference 55). Store and Forwarding switching allows a packet to route next router only after storing entire packet into the memory of current router. In Virtual Cut Through process, forwarding of packet may begin before storing entire packet to the memory of current router to accelerate the packet forwarding process. Packet forwarding in Wormhole switching is slightly different from above two. Here, a packet gets split into numbers of smaller equal sized units called flits. The header flit contains the routing information and remaining flits of that packet follow the same path as it has been followed by the header. Wormhole switching is preferred most for 2D- as well as 3D-based NoC micro-architecture as it consumes low memory area and also provides low latency in cost of increasing deadlock risk.

11.4 Architectural optimization of 3D NoCs

Performance of NoC can be improved significantly optimizing design issues of important micro- and macro-architectural components. Micro-architectural challenges have been described briefly in Section 11.2.1. While macro-architecture comprises study and design of network topologies including the vertical interconnection link among different layers of 3D-based architecture. Design of this vertical interconnect accesses (via) [56] is one of the most important aspects of 3D NoC design, which falls into the overlap between micro- and macro-architectures, as we will see shortly.

11.4.1 Router architecture

Issues related to router designing are tightly coupled with yielding area related to buffer depth, flit size, complexity of the arbiter, and crossbar unit. All of these are considered as major contributing factors in NoC design optimization issue. Modular design of these component makes them suitable for designing 3D integrated router. Such a 3D design of an NoC router helps in reducing the chip footprint and power consumption, leading to an optimized NoC architecture.

Kim *et al.* have proposed a three-dimensional decomposed router that supports additional vertical TSVs for cost optimization of 3D NoC switches. Significant reduction in area, power budget, and logic complexity has been achieved using proposed modified router that supports a partial crossbar switch instead of using a full 3D crossbar. Router architectures (see Figure 11.4) utilizing TSVs are also investigated in MIRA [57] that are using mesh interconnects. Descriptions of three different types

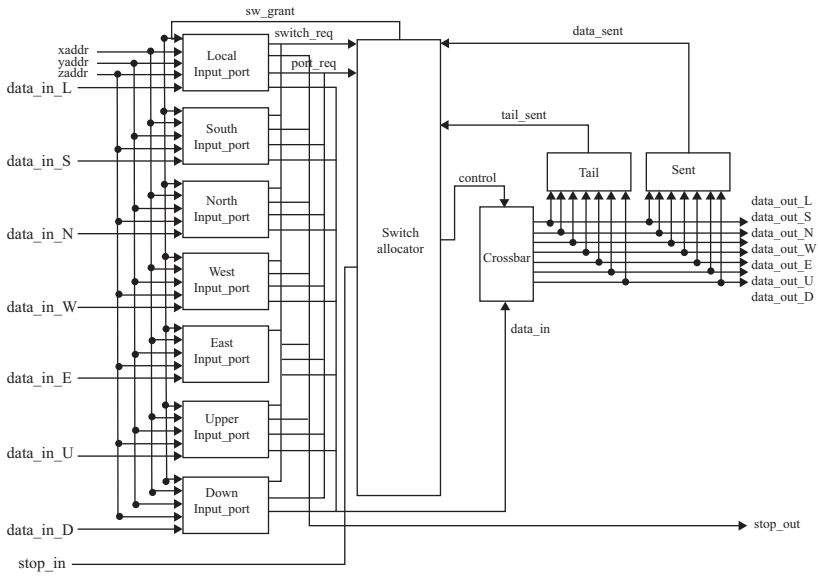


Figure 11.4 NoC router architecture

of 3D-based router design approach namely baseline, multi-layered, and multi-layered with express channel are given there. A baseline router only provides additional port for 3D router design perspective, whereas multi-layered approach allows splitting of router component like input buffer, switching allocator and crossbar, and integration of vertical interconnection (via) among different layers. Routers with high-bandwidth-based express links between non-adjacent routers help in reducing packet latency in cost of additional area overhead.

Performance of this NoC router can be improved through smart pipeline based data forwarding technique [58] and with the help of look-ahead [58] routing and speculative arbitration policy. Some dynamic traffic distribution method [59] can also be applied to reduce the link contention during the switch arbitration time reducing overall latency and consumed power.

Minimizing router input buffer size becomes another important aspect in NoC router optimization issue. This has been achieved through proposed shared buffer design technique in Reference 60, by dynamically varying the number of virtual channels. In Reference 61, a buffer stealing approach has been proposed, which enables one of the router input channels not having sufficient buffer space to utilize unused buffer from other input channel at runtime dynamically. This approaches help in improving router buffer utilization by significantly reducing area requirement without degrading required performance.

11.4.2 Network interface controller

NIC is used to decouple computation resources like intellectual processing cores from the communication network and maintain synchronization between them. The

fundamental function of NIs is to transmit data between the communicating cores through the network infrastructure. One of the important tasks of NIs is to translate the protocol between the PEs and router based on a standard communication protocol such as AXI [62], OCP [63], [64], and DTL [65]. This provides a backward compatibility with the existing IP cores having a distinct interface protocol with respect to the connected network and allows them to be independent of each other. Thus it offers reuse of both, core and communication infrastructure [66] in the existing system. Managing network protocol that includes assembling and splitting of packet, buffer reordering, protocol synchronization, transmitting and receiving of packets, all these are considered as other important tasks to be performed by NIC. To increase parallelism on memory access in network-based multi-processor architectures, adaptive NI architectures are proposed in Reference 67, which use multiple SDRAMs that can be accessed simultaneously. In addition, a smart memory controller unit is implemented in the adaptive NI to improve the memory utilization and to reduce both memory and network latencies. An adaptive memory scheduler which tracks the access patterns of recently scheduled accesses and selects memory accesses matching the pattern of requests is proposed in Reference 68. In such architectures, not only resource utilization and latency are the critical issues but also reordering mechanism is required to deliver the response transactions of concurrent memory accesses in-order. To cope with these issues, an adaptive on-chip NI architecture is presented in Reference 69. The proposed NI exploits an efficient reordering mechanism to handle the in-order delivery and utilizes the AXI transaction-based protocol to bring compatibility with existing IP cores.

11.4.3 Interconnection

In a 3D die-stacking technology, multiple device layers are connected via vertical interconnects tunnelling [70] through them. Different type of vertical interconnect technologies have been evolved, viz. wire bonding, micro-bump, contactless coupling, and TSV vertical interconnect [71]. Among all this assembling technology, TSV approach has become most popular and accepted by 3D integration R&D community section as it offers short and low-loss electrical links with smaller footprint and higher interconnection density. Wafers in this stacked die may be connected either Face-to-Face or Face-to-Back. Face-to-Back approach with TSV interconnects is more scalable when more than two active layers are used. The movement towards a 3D integration design offers increased bandwidth [72] and reduced average interconnection wire length [73], leads to a saving in overall consumed power and also improves system reliability [74]. However, the adoption of 3D integration technology faces the challenges of increasing chip temperature due to increasing power density compared to a planar 2D design. This increased temperature in 3D chips imposes negative impact on system performance by increasing leakage power and cooling cost, leads to generation of hotspot on chip surface. To mitigate such hotspot generation problem, the layout and floor planning of 3D chips should be carefully designed. To address these thermal challenges, several approaches, such as design optimization through intelligent placement [75], insertion of thermal vias [76], [56], and use of novel micro-fluidic cooling polices have been suggested in Reference 77.

11.4.4 Memory

Along with PE, memory is also considered as an important component in development of multi-core SoC system. Optimizing this memory size has important role in designing SoC system that requires large memory storage and memory accesses unit into the chip layout. Applications like audio and video data processing normally deal with this bulk amount of data. Research predicts that memory will consume around two third or even more chip area in future SoC [78] system, making memory an important component for the NoC design, which in turn also determines the cost of the chip. Several research efforts are being carried out on hierarchical organization of memory, data storage strategies, and other possible memory optimization technique. Memory allocation policy and memory packing techniques have been proposed by researchers to minimize cost function driven by memory. This memory allocation is the process of mapping behavioural variables into physical memory. Where initial memory allocation process targets an individual register and register files whereas later efforts address the storage of variables in multi-port memory [79], [80]. In Reference 81, a memory allocation technique for video image-processing systems has been proposed. Besides this memory allocation technique, memory packing [82] is also required when an embedded system is composed of smaller units from a library of components. This packing technique describes the organization of smaller physical memory modules to realize the required logical memory for an application.

Memory subsystem used in NoC architecture is inherently distributed and logically shared over the 3D chip area and usually follows a NUMA (non-uniform memory access) paradigm. With this shared memory space, the memory model is also composed of private memory spaces associated with each PE. The shared address space should maintain a synchronization mechanism in order to exchange data between PEs. In Reference 83, this physically distributed shared memory is dynamically managed by an on-chip hardware memory management unit (HwMMU). Organization of this memory on stacked 3D chip is being categorized as Dance-hall, Sandwich, Per-layer, Terminal, and Mixed architecture in Reference 84, based on layer-wise placement of memories and processors.

11.5 Thermal-aware design

3D die stacking with vertical interconnection technology brings notable improvement over 2D NoC by minimizing the length of the global long interconnection wire, thereby decreasing the packet delay and overall chip area. But in 3D ICs have some drawback due to the higher power density and longer heat dissipation path. Even with balanced traffic distribution, the temperature distribution is not balanced over the chip surface. However, heat generated by inter-active layer might cause a large temperature gradient that leads to generation of hotspot region within the chip area. With a given temperature limitation which can never be exceeded, such unbalanced temperature limits the network performance. To balance temperature distribution, various thermal management techniques, such as physical placement [85] and floor planning [86], use

of thermal vias [87], and micro-fluidic cooling technique [88], [89] has been proposed recently. Distribution of traffic has great impact on generated chip temperature. A traffic-thermal mutual coupling-based co-simulation platform has been described in Reference 21. Router micro-architecture and thermal-aware routing policy [22], [23] may also achieve success on generating low thermal dissipation. Besides this thermal-aware hardware design, temperature management with software assistance [24], [25] especially OS-level task scheduling is becoming hot for multi-core processors based system in recent years.

11.6 Photonic 3D NoC

The objective of this section is to provide a gentle introduction to photonic interconnect and also to review some of the existing state of the art in this field. Readers familiar with PNoCs may skip the following introductory section and jump directly to second subsection.

11.6.1 Photonic interconnect for manycore ICs

MPSoCs scaling to tens or hundreds of PEs are good candidates for PNoCs. This is mainly because such systems exchange large data (usually entire cache lines) between PEs. Photonics has the potential to deliver data at not only high speed but also low power over large intra- and inter-chip distances. Large-scale multi-core ICs have large dies (typically 400 mm^2) and copper-based global interconnect may not satisfy energy-delay product of high-speed data transfers among PEs.

A brief introduction is provided to popular photonic devices and some state-of-the-art PNoCs.

11.6.1.1 Photonic devices and systems

Key components of a photonic link may be broadly classified as follows:

- **Optical source:** This is typically a multi-wavelength LASER source or more recently, LEDs.
- **Optical modulator:** This component modulates a particular wavelength of light with an electrical signal and is an active component.
- **Optical link:** This is a passive multi-mode waveguide capable of carrying multiple wavelengths of light.
- **Optical filter:** Another passive component that selectively extracts a particular wavelength of light from the waveguide.
- **Optical detector:** This active device converts filtered light to an electrical signal.

Silicon Photonics is a nascent field, especially because silicon is an indirect bandgap semiconductor and lasing is difficult on-chip. However a number of

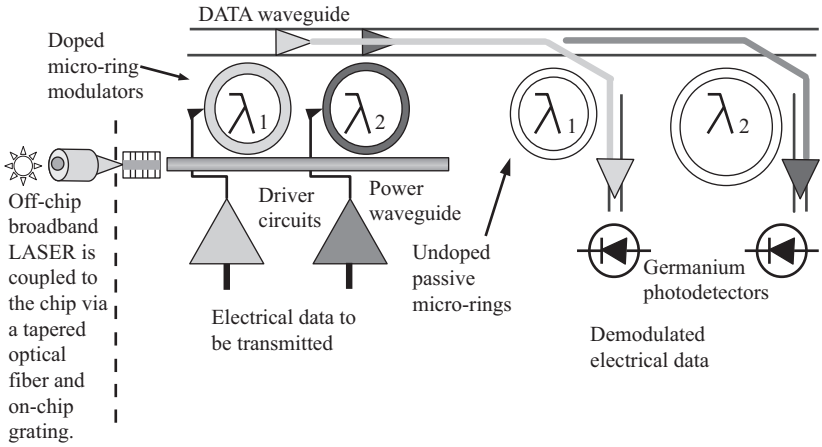


Figure 11.5 Micro-ring-based photonic interconnect

works have explored broadband on-chip LASER sources made from other materials [90]–[93].

Depending on the type of technology used, photonic links are of three types:

- **Micro-ring resonator based:** These commonly used interconnects are based on evanescent coupling of light [94] from one waveguide to a micro-ring or vice versa (see Figure 11.5). A broadband off-chip LASER injects multiple wavelengths of light (usually in the infrared range of 1500–1550 nm) into an *optical fibre*. This fibre is then aligned either with the edge of the chip (lateral coupling) or top of the chip (vertical coupling). At the edge or at the top, there is a diffraction grating that separates the multiple wavelengths of light arriving from the fibre and injects them into the multi-mode waveguide.

  Micro-ring modulators that are doped active PIN-diode devices covert data arriving from a source PE into wavelength specific on–off light signals. Several such wavelengths traverse the wavelength division multiplexed (WDM) waveguide and arrive at the receiving PE after extraction by an undoped micro-ring filter and successive conversion to an electrical signal by germanium photodiodes.
- **Quantum dot (QD) LED based:** QD LEDs are used (rather than on- or off-chip LASERs) that emit light in the visible wavelength range. These devices are much smaller than LASERs, have good stability, and are more energy efficient. As shown in Figure 11.6, a QD LED injects light into a silica (SiO_2) waveguide of about 50 nm diameter [95]. In comparison, polysilicon waveguides used for micro-ring-based interconnect are 500 nm in diameter although they support more wavelengths than their silica counterparts (mainly because of receiver technology). Modulation of the light is done by controlling the current supplied to the QD LED. Due to linear operation of QD LEDs, such a direct scheme works easily.

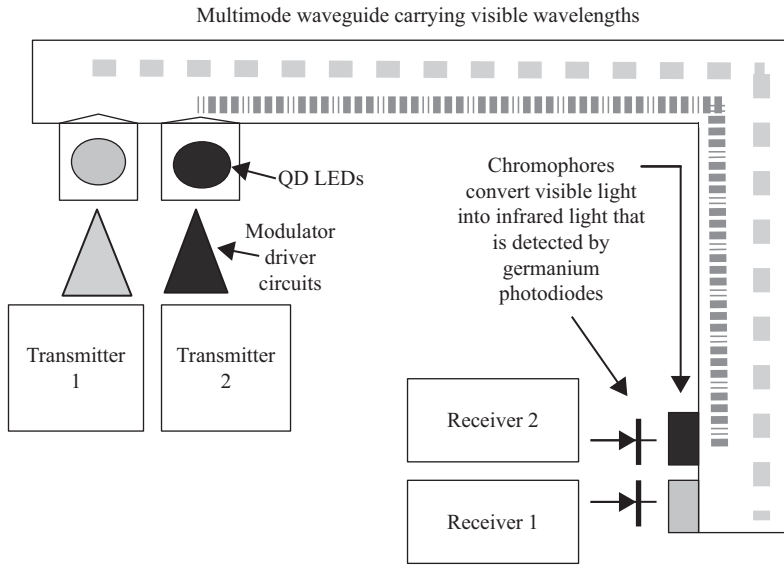


Figure 11.6 QD LED-based photonic interconnect

At the receiving end, optical to electrical conversion must be done via a photodiode. However, germanium photodiodes operate at infrared wavelengths. So, chromophores (special kind of molecules) are used that convert light at visible wavelengths to infrared. Then detection is done. However spacing among the wavelengths must be high, in order to ensure proper detection, and thus the number of wavelength channels is limited to 10 while constructing PNoCs using the above devices.

An NoC utilizing this structure uses a high radix photonic crossbar to connect PEs together. Details of such a scheme are given in Reference 96. Some of the arbitration schemes are discussed later.

- **Free space optical link based:** This is yet another class of technology available for use, especially in 3D NoC. The key idea here is to send data by modulating vertical cavity surface emitting LASER (VCSEL) diodes. Light from these LASERs are allowed to propagate in free space until they strike a reflective surface and are directed to the corresponding photodiode after reflection.

  This technology allows highest transfer rate, but imposes restrictions not only on energy, but also placement of these devices. Fabrication cost is higher and only point-to-point links are supported.

11.6.1.2 Systems based on PNoC

A popular design method for PNoC is to use photonic and electrical interconnect together. Photonic interconnect is well suited for global data transfer with high

bandwidth and low energy. Electrical interconnect is suited for short to medium data transfer whereas it consumes high energy for global interconnect. Such a NoC is called *Hybrid Photonic NoC*. These NoCs have an underlying electrical network and another photonic network that carries large volumes of data from one PE to another optically. When a PE decides to send data to another, it asserts some signals to a path setup controller (centralized or distributed). By sending short path setup messages, an optical circuit or path is built from source to destination. After successful delivery of data optical path is deallocated using path *tear-down* packets.

Shacham *et al.* [97] have shown that topologies like *Torus* and *Folded Torus* can be realized using photonic interconnect with an underlying electrical topology of the same type. Each router in the hybrid network consists of an electrical part and a photonic collision-free switch (comprised of several photonic crossings and bends). Zhang *et al.* [98] have proposed a mesh-based PNoC with special routers where path setup is done purely via photonic signals. Ye *et al.* [99] have proposed a hierarchical torus-based hybrid topology with low latency control-protocols and adaptive power control mechanism. Koohi *et al.* [100] have proposed PeSWAN, a Spidergon photonic topology with separate control plane and optical flow control.

Bahmani *et al.* [101] have designed Corona that has 64 clusters that communicate through an optical crossbar and occasionally an optical broadcast ring. The crossbar and broadcast bus both require a conflict resolution scheme to prevent multiple sources from concurrently sending to the same destination. The arbitration scheme is token based. Pan *et al.* [102] have proposed Firefly, a hybrid hierarchical topology that utilizes photonic reservation assisted Single Write Multiple Read (SWMR) crossbars for global or large data traffic and uses electrical signalling for local or small data communication. *CMesh* topology is used with a concentration of four PEs per router. Joshi *et al.* [103] have shown that a photonic clos topology results in high wavelength utilization but results in higher overall latency. Li *et al.* [104] have proposed LumiNoC, a photonic multi-stage NoC that uses photonic distributed arbitration and achieves lower latencies.

Kurian *et al.* [105] have proposed ATAC, a thousand core photonic ring-based NoC, that uses SWMR links for broadcast traffic and a novel cache coherence protocol and modified directory structure. Grani and Bartolini [106] have proposed a single waveguide ring topology with 68 WDM channels and token-based arbitration. Their design needs much lesser energy compared to ATAC (due to just one data waveguide) however broadcasts are time consuming. Pasricha and Bahirat [107] have proposed OPAL, a 3D hybrid photonic and electrical NoC with dynamic reconfiguration based on changing traffic patterns.

Ciftcioglu *et al.* [108] proposed VCSEL-based free space one hop photonic interconnect for intra-chip networks that has about 4× higher bandwidth than micro-ring resonators. Abousamra *et al.* [109] presented two hop free space interconnect that reduced the number of VCSELs required. Bahirat and Pasricha [110] replaced the fixed two hop interconnect with more energy efficient multiple quantum well LASERs and photodiodes. They have also presented a novel routing, flow control, and collision mitigation scheme for application-specific 3D PNoC.

11.6.2 Multi-dimensional design issues in 3D PNoC

Major system level issues of 3D photonic systems are achieving low power optical networks on the one hand and designing efficient communication protocols on the other hand. If the system is a 3D multi-core, it is imperative to design good external memory access schemes and cache coherent protocols to harness the full optical bandwidth available. Equally important is the optical arbitration technique which decides the time interval of transmissions from multiple sources and destinations.

Let us take up an example 3D system to better understand these concepts. Consider a multi-core computing system that is supposed to run parallel workloads e.g. SPLASH 2 suite [111]. For the sake of brevity, we assume that this chip will run only one benchmark at a given time. We arbitrarily choose 64 PEs (to facilitate use of a vanilla FullMap Directory) and the medium working set. With data collected from the suite specification, for an L1 cache size of 32 KB (4 way) and L2 cache size of 512 KB (8 way), the system performs quite well for memory intensive benchmarks. So the cache line size is 64 Bytes. A photonic message must be at least this size.

For above system if we consider more than one tier (or 3D layers), multiple arrangements are possible depending on whether a PE includes the cache memory or not. For 2D systems, it is better to consider that a PE includes both processor and memory. However for thermal issues, we consider a single processor layer (64 PEs) and a memory layer on top of the processor layer. The PNoC is placed at the topmost layer. A heat sink with heat-spreader is mounted on top of the chip. Inter-layer data communication is accomplished with TSVs. If the fabrication technology permits more layers, we can reduce the number of processors per layer, and hence improve on-chip latency and power at the cost of higher heat dissipation per unit area. Thermal vias are special TSVs that carry the heat away from hotspots.

Once the number of layers are known beforehand, we can design the interconnect based on these guidelines. First an appropriate topology must be selected that maximizes throughput and lowers energy. For PNoC, an on- or off-chip LASER injects several wavelengths into a multi-mode waveguide. The number of wavelengths depends on the LASER power and insertion loss of the optical network components. As stated in the previous section, several photonic topologies and routers exist each having its own merits and demerits.

A very important problem that PNoC designers must face is the issue of multicast. Multi-casts arise in all multi-core systems due to cache invalidations. Multi-cast with several destinations necessitates the role of arbitration in a dense WDM NoC. Each wavelength may be shared by several sources, and in such systems arbitration is all the more important. Arbitration may be distributed (token based) where a small photonic message (a single word) circulates around the network. Interested senders may grab this token, and the one that wins this arbitration gets to transmit to the destinations. Flow control may be considered, given the fact that buffer space is limited.

For our example system, a photonic network that can serve the bandwidth requirement of 64 Bytes per transfer is required. Latency and energy are important factors to consider while designing the topology. Optical Crosstalk and Loss Analysis Platform,

or CLAP [112], may be used to obtain some of the topology parameters like signal-to-noise ratio. After initial topology is decided it is important to simulate the entire system on a cycle accurate [113] or dynamic instrumentation-based simulator [114].

11.7 Wireless 3D NoC

3D ICs mitigate the problems of skew, jitter, and delay caused by scaling of metal interconnects in planar 2D ICs. Here, TSVs are commonly selected as communication medium between the tiers of the 3D IC stack. But, TSVs suffer from a considerable utilization of the wiring footprint of the individual tiers leaving less area for intra-tier routing and device placement, thus imposes constraints on the number of TSVs per unit area [115]. Moreover, the process of fabricating TSVs for a multi-tier (more than two tiers) interconnection on a 3D IC is difficult for some TSV technologies and entirely prohibited for certain others [115]. That is intra-tier routing is necessary to communicate separated communication end-points on two separate tiers. This communication challenge can be overcome by the use of RF-based interconnect medium.

11.7.1 Low-latency-based wireless 3D NoCs

A wireless 3D IC system is a collection of dies, each of which is developed more or less independently. Such dies built upon different process technologies, for example, processor die and DRAM die, designed independently and stacked later only. Adding randomly wired shortcut NoCs to wireless 3D systems ensures a balance between modular design and minimum design complexity for low-latency-based system.

Low latency wireless 3D NoCs via randomized shortcut chips are introduced in Reference 116. It reduces communication latency while incurring minimum design complexity by using random connectivity via wireless 3D NoCs. Two different cases are considered here: (i) replacing the existing horizontal 2D NoCs in a wireless 3D NoC with random shortcut NoCs and (ii) adding a random NoC chip, in which the horizontally wired links are randomly determined, to a wireless 3D platform with partial or no horizontal NoCs in order to achieve full connectivity. Figure11.7 shows the latter case.

A random NoC chip can be built with a unique horizontal wiring pattern by reconfiguring the switch boxes randomly which remove redundant horizontal links (in addition to the regular links) connected via routers and FPGA-like switch boxes on the same die. In this platform, the packets routing can be optimized by utilizing both the newly added random NoCs and the existing regular NoCs by using the irregular up or down routing [117] policy. Small-world wireless 2D NoCs that employ millimetre-wave have been recently used in Reference 118. A runtime macro-scale topology reconfiguration, called skip-links, has been also proposed to dynamically utilize the small-world effects of on-chip networks [119].

Inductive coupling [34], [120], [121] is a die-level wireless interconnection scheme that uses square coils as data transmitters. The coils can be implemented

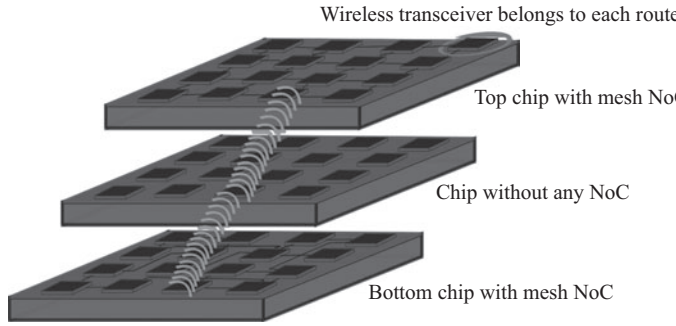


Figure 11.7 Wireless 3D NoC

with common metal layers of the chip; thus no special process technology is required. This coupling can enable the customization of hardware components (or chips) in a package to satisfy the application requirements at a low cost. That is, we can add, remove, and swap the known-good-dies (e.g. processor and memory chips) in a package to meet the requirements without making new mask patterns. Adding random connectivity via wireless 3D NoCs can reduce communication latency with minimum design complexity. Both vertical buses and P2P [122] links are implemented with the inductive coupling, and either one can be used depending on the number of chips stacked and communication pattern, such as unicast- or multi-cast-based communication.The random topology chip is efficient for various combinations of non-random topology chips for 3D NoCs.

3D NoCs that include random topology use an irregular routing algorithm, such as up or down routing [117]. Similar type routing strategy is also used in Reference 123, to efficiently route packets in 3D wireless NoCs. By this strategy the best spanning tree root that can minimize the hop count is selected for each message class. Then, routing paths are generated under the up or down rule with the selected spanning tree roots. Using this routing strategy, the packet routing can be optimized by exploiting both the existing regular NoCs and the newly added random NoCs.

The application execution time results also reflect the communication latency reduction. This can prove beneficial for future complex heterogeneous computing platforms where deterministic and regularity of nodes may be lost, and average flit transfer energy can also be improved accordingly. Finally it can be said adding random NoC chips to a wireless 3D system strikes a good balance between the modular design and low latency centric design.

11.7.2 Inductive coupling interconnected application-specific 3D NoC

Wireless interconnects in 3D NoC architecture [124] for application-specific SoC design can be implemented using capacitively coupled [125] and inductively

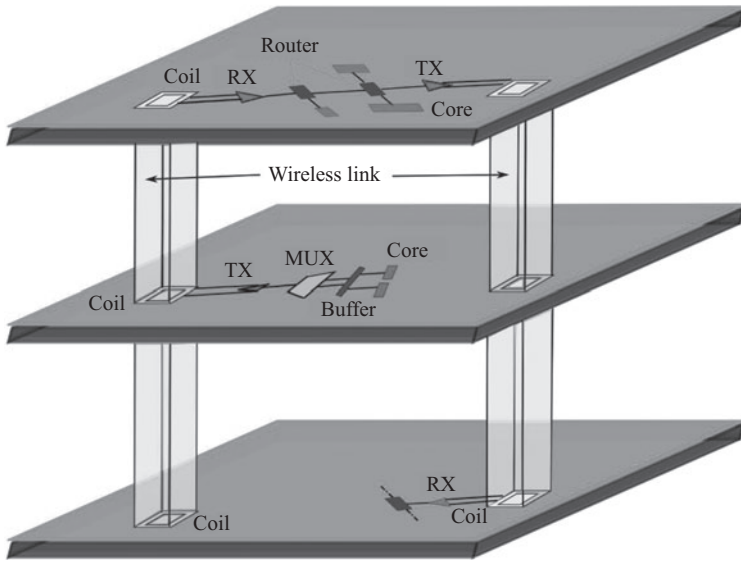


Figure 11.8 Inductively coupled wireless 3D NoC architecture

coupled [126] interconnects as shown in Figure 11.8 instead of TSV used for inter-layer communication (see Figure 11.6). It cuts down manufacturing cost, improves interconnect performance, reduces power consumption, and provides larger design space. The key to this design is allocating wireless links in 3D on-chip network effectively while maintaining signal integrity.

Several issues need to be considered before replacing TSV interconnects with inductively coupled interconnects, viz. signal interference between coils of adjacent channels and efficient allocation of wireless interconnections as a result of relatively large channel pitch of wireless links. A time-division multiplexing (TDM) protocol is used to share the channel between several tiers. This architecture can solve interference problem by fixing transceiver locations and using TDM protocol. It can be used in homogeneous SoC design and successfully avoids the design complexity brought by channel interference and improves channel efficiency. It also improves overall system performance providing low latency, higher throughput with significant reduction in consumed energy.

11.7.3 Reconfigurable hybrid 3D wireless NoC

In the RF interconnect based NoCs, the data are transmitted from one communication end-point to the other using EM waves. There are two types of RF IC interconnects of node: The micro-strip transmission line based interconnects operating in the RF range [127] and the wireless communication based intra-chip interconnects operating in the RF range [128]. This hybrid 3D wireless NoC integrates the wireless RF interconnects

with traditional metal wire based interconnects. The network throughput and latency improvements using the hybrid wireless NoC (WiNoC) are shown in References 33, 129, and 130. The concept of the hybrid wireless NoCs is applied here to 3D NoCs making it a 3D WiNoC.

11.8 3D NoC simulators

Network simulation forms a quintessential part of network design. A designer designs a network keeping in mind several functional and quality of service parameters. A good network simulator fulfils the following goals:

- **Functional verification:** Does the designed network comply with desired behaviour. Do all protocols used in the design function according to specifications.
- **Controllability of simulation parameters:** Those parameters that may have a continuum of values need to be properly swept from their lower limits to higher limits. Various parameters may not be compatible with each other and may have several dependencies. Designers should be able to tweak and specify the complex relationships among such conflicting parameters with ease.
- **Observability of simulation parameters:** Results of the simulation must be presented in a lucid but informative way to the designer (often using graphic aids). Data interpretation and analysis tools must be powerful enough to be used at abstract levels as well as provide insight into both fine grained data and control behaviour.
- **Design closure at higher speed:** This is the most important point in current scenarios where a design needs to achieve closure with as little effort and iterations as possible. Design closure must maintain a customizable balance between speed and accuracy. We will elaborate on this issue in the next section as it is especially important to achieve a small turnaround time with least human intervention in NoC simulators.

11.8.1 NoC simulation

In previous section we discussed about general network simulators. In this section we focus mainly on two things, salient features of a good NoC simulator and more importantly, how these features may have different implications for different simulation scenarios.

We would like to point out that several different scenarios may need multiple aspects of the same or different simulators (because one simulator may be aimed at testing just one aspect while another one at some other design aspect). As we will soon see, a common framework with modular architecture may help in achieving design closure at sufficient accuracy and speed.

Salient features of a good NoC simulator are stated (but not limited to) below.

- **Specification entry:** Once a designer collects relevant information about the intended design of the NoC, he or she must be able to formulate a suitable abstract specification of the design so that the simulation software may be able to verify the design. There are several ways to do this:
  1. Hardware Specification Languages (like VHDL, Verilog, System Verilog [131])
  2. Specific Libraries and associated Application Programming Interfaces (like SytemC [132])
  3. Scripting Languages (like Tcl/Tk, Perl, Python)
  4. Custom or Tool Specific Languages (like NED in the tool OMNet++ [133])
  5. Graphical User Interfaces (like Platform Architect [134] from Synopsys, DesignPlayer [135])
- **Formal methods used to model system behaviour:** In order to model traffic and timing behaviour of NoCs, deterministic finite state machine based models are too simple as a model of communication. So several probabilistic models have been researched to accurately model packet traffic and transmission channels. Queuing theory [136] and network calculus [137] are widely used theories to build such models.

  While most queuing models are based on Markovian statistics, there are some works on Markov modulated Poisson processes (MMPP) [138]. Most simulators use M/G/1 queues to model packet contention at routers. However there are some shortcomings of this approach, most important being the difficulty to model virtual channels, faulty links and self-similar traffic. The latter is receiving significant research attention. Fractal traffic patterns and their modelling using Hurst Parameters is an important area of study.

  Network calculus with min-plus algebra is used to estimate arrival and service curves for a particular network. Several important properties and trends like skew and jitter of traffic may be accurately captured using such mathematical tools.
- **Detailed timing behaviour:** There are broadly three classes of timing models available for simulators:
  - **Event-driven simulation:** Vast majority of simulators are variants of this model. In such a model, all changes occurring while the network is simulated are some form of event e.g. arrival of packets at a queue, requests arriving at the home tile of a tiled multi-core chip, etc.

    There is an event stack that is shared by all event generators. An event with an associated timestamp (usually a 64 bit integer value) is pushed into the stack when it occurs. After all events prior to current one are served, current event is popped and checked if some new events need to be generated. Correspondingly, these new events are again pushed into the stack.

    The stack may be implemented as a priority queue with timestamp as priority information. Time advances when an event is processed and never between events. Simulation stops either when an event's timestamp exceeds the user-defined time limit or when no more events remain in the stack.

OMNet++ [133] and SystemC [132] based simulation are examples of this methodology.

– **Cycle accurate simulation:** In this methodology concurrent behaviour of actual hardware may be simulated by using Hardware Description Languages. Low-level RTL may be easily simulated using this approach. Several good texts are already available that discuss this methodology.

  However the only and major drawback of this method is the high time required to develop and debug new hardware modules. Sometimes it is also necessary to provide abstract views to designers regarding their design's overall system behaviour (like contention and throughput) which becomes difficult at individual cycle levels. Also, design closure is difficult, error-prone and time consuming in such simulation environments and emulation on Gate Arrays are sometimes required to study exact real-world traffic.

  Inspite of its disadvantages this method is still quite popular in industrial settings where there are several in-house or third-party IPs available and reliability of the design is the most important issue.

– **Cycle specific simulation:** This is a hybrid methodology that is a combination of both cycle accurate and event-driven simulation. Such a method is gaining acceptance from both industry and research circles due to its ability to model both hardware and software. Various algorithms are used to get the best of both practices. Manifold [113] is such a simulator in the open source domain.

- **Physical models to evaluate power and area:** Accurate power and energy models need to be present so that system and network energy may be calculated. In general, energy spent to do some processing or link traversal are stored as a database and various events retrieve the energy and calculate power at various time periods using the energy. Area is calculated only once using technology files. Orion [139] and DSENT [140] are energy models for network modules.

  Another approach is to use the activity trace and calculate dynamic power as done in cycle accurate simulation methodology.

- **Hardware-software co-design:** In the current context of multi-core systems on chip, current NoCs have to cater to both fixed function and programmable cores. Real-world applications need a mix of several varieties of PEs like accelerators, reconfigurable hardware, and digital signal processors.

  Therefore, need for a combined approach that models software and hardware on the same simulation platform is inevitable. There are, in general, two approaches for this type of simulation:

  – **Full system simulation:** In this method a virtual machine (VM) like environment is used, along with an operating system kernel. Pre-compiled version of the application to be run on the VM is cross-compiled using a suitable cross compiler.

    For example, if a benchmark application is to be run on an alpha processor, the corresponding virtual machine or emulator environment containing alpha processor instruction set architecture (ISA) is run on the host machine/server and modified linux kernel image is mounted on it. Then the

benchmark/application is executed by a shell script either preloaded into the kernel-image or via terminal using some host TCP/IP port.

The above method is followed by simulators like GEM5 [141] and Simics [142].

– **Binary instrumentation based:** In this method a binary instrumentation tool like Intel's PIN is used to read an application executable compiled for the target architecture. Instrumentation is done in trace mode, where the executable is run as a guest application and the simulator is actually built as a plugin or interface to simulate memory accesses and input/output.

This method is highly scalable (up to 1000 cores) and very fast compared to full system simulation at the expense of slight loss of accuracy and higher complexity. Simulators using this method are Snipersim [143], GraphiteSim [114], Simplescalar [144].

- **Fault models:** Real-world applications and systems very often suffer from faults. Reliability is a major concern for such systems. To test reliability of such systems, fault models such as permanent and transient faults are used. Examples of fault models include link and partial router failure or Through Silicon Via failure. More detailed the failure model better is the resulting fault simulator's quality.

  Often user has to specify fault models and run full system simulation to find out whether the adaptive fault tolerant NoC routing algorithm is functioning reliably. FaultModel [145] is a detailed fault modelling tool which runs with GEM5.
- **Thermal models:** Thermal modelling is mainly required to detect hotspots. Thermal-aware routing algorithms and their effect on such hotspots need to be accurately reported by thermal-aware simulators. A popular thermal simulator is Hotspot [146].
- **DVFS awareness:** Dynamic Voltage and Frequency Scaling is very important in today's nanoscale design environment due to leakage effects and corresponding power loss. The simulator should be able to vary the speed of operation and also to report the effect on power consumption due to DVFS.
- **3D IC support:** 3D NoC support is a new requirement and usually any general purpose simulator may be extended to support 3D models. Some of the major concerns is heat removal. 3D-ICE [147] is an open tool that is devoted to cooling solutions, especially micro-fluidic channels.

11.9 Reliability and fault tolerance in 3D NoCs

Rapid technology scaling in deep sub-micron domain has exacerbated the reliability issues in chip interconnects scenario. During the chip manufacturing process at deep nanometre technologies (65 nm and beyond), some variations like random dopant fluctuations (RDF), overlay and spatial-correlated effects like dose, focus, etc. are almost inevitable. These in turn impact on changing device characteristics, like effective gate length, oxide thickness, transistor threshold voltages leading to violation in consumed power and timing constraints and induced hazard in the system performance, raise the

reliability issues. Hard faults are commonly generated from ageing effects, such as electromigration and others manufacturing errors and testing process. Whereas soft errors are usually caused from crosstalk, coupling noise and others transient effect. Depending on duration of time a fault persists in a real-time system can be categorized in three different types depending on its manifestation pattern: transient, intermittent, and permanent faults.

Transient faults are temporary and relate to soft errors that usually lead to Single Event Upsets (SEUs) by inverting state of the transistor or other storage cells. This type of faults is random in nature and therefore hard to predict. A common measurement for transient fault is Soft Error Rate (SER) that describes the probability of occurrence of error in the circuitry and tolerance of the circuit against variable phenomena (like electromigration radiation and other noises originating from other parts of the chip) causing soft error. SER can be decreased by giving special care to low noise properties at the circuit design time.

While intermittent faults arise in the system due to timing discrepancies like gate and path delay, the circuit is operable but after some alteration in environmental conditions such as change in temperature or voltage that violets the operation. Intermittent fault often leads to occurrence of permanent failure. It is very hard to detect these faults, because most of the time system operates correctly and gets failure only under certain input condition or for some specific input instances. The way to recover this fault is to replace the faulty circuit or to bypass the faulty region.

Failure of router intra-links and failure of router components like routing computation unit, input buffer, virtual channel arbiter, switching allocator and crossbar unit, all these are susceptible to different types of permanent faults. Permanent fault causes irreversible physical changes in the circuit and therefore they are easily traceable.

A reliable NoC system should detect occurrence of all these different kinds of faults and then working on reconfiguring the system resources to recover these faults and ensuring guarantees to the continuous correct functionality of the system. Presence of these faults is detected by custom testing mechanisms or other detection schemes based on codes. Codes are mostly used to detect and correct errors of transient faults. For instance, Crosstalk Avoidance Codes (CAC) [148] are used to avoid crosstalk and minimize the chance of occurring noise in the transmitting wire. For errors whose presence could not be detected, a Error Detection Code (EDC) is followed by Error Correction Code (ECC) [149] to detect and correct these errors.

Retransmission scheme can solve the transient fault (usually caused from crosstalk) but not the intermittent and permanent faults. A testing process is carried out to detect any kind of incompetency that are induced into the chip during manufacturing time. While adaptive routing is used to bypass the defective region and to tackle the permanent faults that occur during operational time. However, deadlock is one important issue that may arise with adaptive routing. Most of the existing 3D NoC routing algorithms use turn-model that imposes restriction on available routing choices. AdaptiveZ [53] and Adaptive XYZ [44], [54] are some of the recently proposed fault tolerant adaptive routing for 3D mesh NoC. While some technique uses Virtual-Channel (VC) [150] for deadlock-avoidance others employ Virtual-Output-Queue (VOQ) [151]. All these procedures ensure the deadlock freeness at the expense

of additional hardware cost, increasing implementation complexity, or from latency penalty caused by direction restricted routing.

The reconfigurable approach proposed in Reference 152 tolerates different kind of faults at runtime, where predetermined Hamming coding and interleaving combined with stop-and-wait ARQ to deal with transient, intermittent, and permanent faults that occur as both bursts and single errors. Various error control schemes are provided in References 153 and 154 to achieve the required QoS levels.

11.10 Conclusion

With continuous and rapid progress in different emerging technologies and design techniques development of 3D NoC is indeed a fast growing field and grabbed the attention of a majority of serious researchers. Besides development in different 3D integration technologies development in several emerging technologies like photonic interconnects for very high bandwidth data transfer, emergence, and successful application of wireless communication technologies in nanoscale domains, growth in other interconnect technologies like CNT, graphene, etc. are continuously working in harmony towards the rapid development of high-performance 3D NoCs for different high-performance computing and communication intensive applications. Several open challenges still exist in this domain like *Development of Novel Nanoscale Technologies for Photonics* where Photonic crystal based interconnect [155] may be considered for constructing 3D NoC due to their high confinement capabilities. These interconnects have extremely low loss and allow perpendicular bends. Another direction for *Novel Architectures and System Development* may also be mentioned in this regard. Photonics will soon be a key enabler of on-chip exascale computing. General Purpose Graphics Processing Units (GPGPUs) and low power photonic interconnect for such systems are being considered now. Proper dimensional scaling without significant loss in high throughput and performance always remains a challenging issue for wireless and other emerging technologies. As an end note, ultra high-performance exascale computing is just going to be a reality in few years with enormous possibilities of 3D NoCs with proper support of other emerging technologies discussed above.

References

[1] S. J. Eggers, J. S. Emer, H. M. Leby, *et al.*, "Simultaneous multithreading: a platform for next-generation processors," *Micro, IEEE*, vol. 17, no. 5, pp. 12–19, 1997.

[2] K. Olukotun, "A single-chip multiprocessor," *Computer*, vol. 30, no. 9, pp. 79–85, Sep. 1997.

[3] W. Wolf, A. A. Jerraya, and G. Martin, "Multiprocessor system-on-chip (mpsoc) technology," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 10, pp. 1701–1713, 2008.

[4] J. Cong, "An interconnect-centric design flow for nanometer technologies," *Proceedings of the IEEE*, vol. 89, no. 4, pp. 505–528, 2001.

[5] J. P. Ville Rantala and T. Lehtonen, "Network on chip routing algorithms. TUCS technical reports 779," Turku Centre for Computer Science, Tech. Rep., 2006.

[6] J. Kim, C. Nicopoulos, D. Park, *et al.*, "A novel dimensionally-decomposed router for on-chip communication in 3D architectures," *SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 138–149, Jun. 2007.

[7] K. Puttaswamy and G. Loh, "Thermal herding: Microarchitecture techniques for controlling hotspots in high-performance 3D-integrated processors," in *IEEE 13th International Symposium on High Performance Computer Architecture, 2007. HPCA 2007*, pp. 193–204, Feb. 2007.

[8] L. Shang, L.-S. Peh, A. Kumar, and N. Jha, "Thermal modeling, characterization and management of on-chip networks," in *37th International Symposium on Microarchitecture, 2004. MICRO-37 2004*, pp. 67–78, Dec. 2004.

[9] W. Davis, J. Wilson, S. Mick, *et al.*, "Demystifying 3D ICs: the pros and cons of going vertical," *Design Test of Computers, IEEE*, vol. 22, no. 6, pp. 498–510, Nov. 2005.

[10] L. Benini, "3D-mpsocs: architectural and design technology outlook," in *Keynote Presentation at 7th International Forum on Application Specific MultiProcessor SoC*, 2008.

[11] J. Sepulveda, G. Gogniat, R. Pires, W. Chau, and M. Strum, "An evolutive approach for designing thermal and performance-aware heterogeneous 3D-NoCs," in *2013 26th Symposium on Integrated Circuits and Systems Design (SBCCI)*, pp. 1–6, Sept. 2013.

[12] C. Addo-Quaye, "Thermal-aware mapping and placement for 3-D NoC designs," in *SOC Conference, 2005. Proceedings. IEEE International*, pp. 25–28, Sept. 2005.

[13] K. Siozios, I. Anagnostopoulos, and D. Soudris, "A high-level mapping algorithm targeting 3D NoC architectures with multiple vdd," in *2010 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 444–445, July 2010.

[14] P. Sahu, T. Shah, K. Manna, and S. Chattopadhyay, "Application mapping onto mesh-based network-on-chip using discrete particle swarm optimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 300–312, Feb. 2014.

[15] Y.-R. Huang, J.-H. Pan, and Y.-C. Lu, "Thermal-aware router-sharing architecture for 3D network-on-chip designs," in *2010 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 1087–1090, Dec. 2010.

[16] V. Nandakumar and M. Marek-Sadowska, "A low energy network-on-chip fabric for 3-D multi-core architectures," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 266–277, Jun. 2012.

[17] X.-H. Wang, P. Liu, M. Yang, *et al.*, "Energy efficient run-time incremental mapping for 3-D networks-on-chip," *Journal of Computer Science and Technology*, vol. 28, no. 1, pp. 54–71, 2013.

[18] J. Cong and Y. Zhang, "Thermal via planning for 3-D ICs," in *IEEE/ACM International Conference on Computer-Aided Design, 2005. ICCAD-2005*, pp. 745–752, Nov. 2005.

[19] J. Cong, G. Luo, J. Wei, and Y. Zhang, "Thermal-aware 3D IC placement via transformation," in *Design Automation Conference, 2007. ASP-DAC '07. Asia and South Pacific*, pp. 780–785, Jan. 2007.

[20] T. Zhang, Y. Zhan, and S. Sapatnekar, "Temperature-aware routing in 3D ICs," in *Asia and South Pacific Conference on Design Automation, 2006*, p. 6, Jan. 2006.

[21] K.-Y. Jheng, C.-H. Chao, H.-Y. Wang, and A.-Y. Wu, "Traffic-thermal mutual-coupling co-simulation platform for three-dimensional network-on-chip," in *2010 International Symposium on VLSI Design Automation and Test (VLSI-DAT)*, pp. 135–138, April 2010.

[22] I. Anagnostopoulos, A. Bartzas, and D. Soudris, "Temperature-aware platform optimizations for 2D and 3D networks-on-chip."

[23] M. Daneshtalab, A. Sobhani, A. Afzali-Kusha, O. Fatemi, and Z. Navabi, "NoC hot spot minimization using antnet dynamic routing algorithm," in *International Conference on Application-Specific Systems, Architectures and Processors, 2006. ASAP '06*, pp. 33–38, Sept. 2006.

[24] X. Zhou, J. Yang, Y. Xu, Y. Zhang, and J. Zhao, "Thermal-aware task scheduling for 3D multicore processors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 1, pp. 60–71, Jan. 2010.

[25] A. Coskun, T. Rosing, K. Whisnant, and K. Gross, "Temperature-aware mpsoc scheduling for reducing hot spots and gradients," in *Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific*, pp. 49–54, Mar. 2008.

[26] T. Lehtonen, P. Liljeberg, and J. Plosila, "Analysis of forward error correction methods for nanoscale networks-on-chip," in *Proceedings of the 2Nd International Conference on Nano-Networks*, ICST, Brussels, Belgium, pp. 3:1–3:5, 2007.

[27] M. Hsiao, "A class of optimal minimum odd-weight-column sec-ded codes," *IBM Journal of Research and Development*, vol. 14, no. 4, pp. 395–401, Jul. 1970.

[28] R. Saleh, S. Wilton, S. Mirabbasi, *et al.*, "System-on-chip: reuse and integration," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1050–1069, Jun. 2006.

[29] K. Kanda, D. Antono, K. Ishida, *et al.*, "1.27gb/s/pin 3mw/pin wireless superconnect (wsc) interface scheme," in *Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC. 2003 IEEE International*, vol. 1, pp. 186–487, Feb. 2003.

[30] N. Miura, T. Sakura, and T. Kuroda, "A 1.2gb/s/pin wireless superconnect based on inductive inter-chip signaling (iis)," in *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, vol. 1, pp. 142–517, Feb. 2004.

[31] A. Rashid, S. Watanabe, and T. Kikkawa, "High transmission gain integrated antenna on extremely high resistivity si for ulsi wireless interconnect," *Electron Device Letters, IEEE*, vol. 23, no. 12, pp. 731–733, Dec. 2002.

[32] M.-C. Chang, V. Roychowdhury, L. Zhang, H. Shin, and Y. Qian, "Rf/wireless interconnect for inter- and intra-chip communications," *Proceedings of the IEEE*, vol. 89, no. 4, pp. 456–466, Apr. 2001.

[33] B. Floyd, C.-M. Hung, and K. O, "Intra-chip wireless interconnect for clock distribution implemented with integrated antennas, receivers, and transmitters," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 5, pp. 543–552, May 2002.

[34] N. Miura, H. Ishikuro, T. Sakurai, and T. Kuroda, "A 0.14pj/b inductive-coupling inter-chip data transceiver with digitally-controlled precise pulse shaping," in *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pp. 358–608, Feb. 2007.

[35] A. Fazzi, R. Canegallo, L. Ciccarelli, *et al.*, "3D capacitive interconnections with mono- and bi-directional capabilities," in *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pp. 356–608, Feb. 2007.

[36] M. Chang, J. Cong, A. Kaplan, *et al.*, "Cmp network-on-chip overlaid with multi-band rf-interconnect," in *IEEE 14th International Symposium on High Performance Computer Architecture, 2008. HPCA 2008*, pp. 191–202, Feb. 2008.

[37] P. Avouris, "Carbon nanotube electronics and photonics," *Physics Today*, vol. 62, no. 1, pp. 34–40, 2009.

[38] F. Rana, "Graphene terahertz plasmon oscillators," *IEEE Transactions on Nanotechnology*, vol. 7, no. 1, pp. 91–99, Jan. 2008.

[39] J. M. Jornet and I. F. Akyildiz, "Graphene-based nano-antennas for electromagnetic nanocommunications in the terahertz band," in *2010 Proceedings of the Fourth European Conference on Antennas and Propagation (EuCAP),* IEEE, pp. 1–5, 2010.

[40] G. T. Reed, Ed., *Silicon Photonics, The State of the Art*, 1st ed. West Sussex, England: John Wiley and Sons, 2008.

[41] Intel-Labs, "The 50g silicon photonics link," Intel, Tech. Rep., 2010. [Online]. Available: http://www.intel.com/content/www/us/en/data-center/silicon-photonics-50g-link-paper.html

[42] I. O'Connor and G. Nicolescu, Eds., *Integrated Optical Interconnect Architectures for Embedded Systems*. New York, NY: Springer, 2013.

[43] W. Zhang, L. Hou, J. Wang, S. Geng, and W. Wu, "Comparison research between XY and odd–even routing algorithm of a 2-dimension 3 × 3 mesh topology network-on-chip," in *IRE Transactions on Electronic Computers*, vol. IEEVOL-3, pp. 329–333, 2009.

[44] A.-M. Rahmani, K. Latif, K. Vaddina, *et al.*, "Arb-net: a novel adaptive monitoring platform for stacked mesh 3D NoC architectures," in *Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, pp. 413–418, Jan. 2012.

[45] M. Bahmani, A. Sheibanyrad, F. Petrot, F. Dubois, and P. Durante, "A 3D-NoC router implementation exploiting vertically-partially-connected topologies," in *2012 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 9–14, Aug. 2012.

[46] H. Matsutani, M. Koibuchi, D. Hsu, and H. Amano, "Three-dimensional layout of on-chip tree-based networks," in *International Symposium on Parallel Architectures, Algorithms, and Networks, 2008. I-SPAN 2008*, pp. 281–288, May 2008.

[47] A. Bose, P. Ghosal, and S. P. Mohanty, "A low latency scalable 3D NoC using bft topology with table based uniform routing," in *Proceedings of the 2014 IEEE Computer Society Annual Symposium on VLSI*, ser. ISVLSI '14. Washington, DC, USA: IEEE Computer Society, pp. 136–141, 2014.

[48] C. Glass and L. Ni, "Adaptive routing in mesh-connected networks," in *Proceedings of the 12th International Conference on Distributed Computing Systems, 1992*, pp. 12–19, Jun. 1992.

[49] C.-H. Chao, K.-Y. Jheng, H.-Y. Wang, J.-C. Wu, and A.-Y. Wu, "Traffic- and thermal-aware run-time thermal management scheme for 3D NoC systems," in *2010 Fourth ACM/IEEE International Symposium on Networks-on-Chip (NoCS)*, pp. 223–230, May 2010.

[50] M. Ahmed and R. Kumar, "Parameterized path-based, randomized, oblivious, minimal routing in 3D mesh NoC," in *TENCON 2012 – 2012 IEEE Region 10 Conference*, pp. 1–6, Nov. 2012.

[51] S.-Y. Lin, T.-C. Yin, H.-Y. Wang, and A.-Y. Wu, "Traffic-and thermal-aware routing for throttled three-dimensional network-on-chip systems," in *2011 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pp. 1–4, Apr. 2011.

[52] R. Al-Dujaily, N. Dahir, T. Mak, F. Xia, and A. Yakovlev, "Dynamic programming-based runtime thermal management (dprtm): an online thermal control strategy for 3D-NoC systems," *ACM Transactions on Design Automation of Electronic Systems*, vol. 19, no. 1, pp. 2:1–2:27, Dec. 2013.

[53] A.-M. Rahmani, P. Liljeberg, K. Latif, *et al.*, "Congestion aware, fault tolerant, and thermally efficient inter-layer communication scheme for hybrid NoC-bus 3D architectures," in *2011 Fifth IEEE/ACM International Symposium on Networks on Chip (NoCS)*, pp. 65–72, May 2011.

[54] C. Feng, M. Zhang, J. Li, *et al.*, "A low-overhead fault-aware deflection routing algorithm for 3D network-on-chip," in *2011 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 19–24, July 2011.

[55] V. Rantala, T. Lehtonen, and J. Plosila, *Network on chip routing algorithms*. Citeseer, 2006.

[56] S. Kumar and R. van Leuken, "A 3D network-on-chip for stacked-die transactional chip multiprocessors using through silicon vias," in *2011 6th International Conference on Design Technology of Integrated Systems in Nanoscale Era (DTIS)*, pp. 1–6, Apr. 2011.

[57] D. Park, S. Eachempati, R. Das, *et al.*, "Mira: a multi-layered on-chip interconnect router architecture," in *35th International Symposium on Computer Architecture, 2008. ISCA '08*, pp. 251–261, Jun. 2008.

[58] A. Ahmed and A. Abdallah, "La-xyz: low latency, high throughput look-ahead routing algorithm for 3D network-on-chip (3D-NoC) architecture," in *2012 IEEE 6th International Symposium on Embedded Multicore Socs (MCSoC)*, pp. 167–174, Sept. 2012.

[59] J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, and C. Das, "A low latency router supporting adaptivity for on-chip interconnects," in *42nd Design Automation Conference, 2005. Proceedings*, pp. 559–564, Jun. 2005.

[60] C. Nicopoulos, D. Park, J. Kim, *et al.*, "Vichar: a dynamic virtual channel regulator for network-on-chip routers," in *39th Annual IEEE/ACM International Symposium on Microarchitecture, 2006. MICRO-39*, pp. 333–346, Dec. 2006.

[61] W.-T. Su, J.-S. Shen, and P.-A. Hsiung, "Network-on-chip router design with buffer-stealing," in *Design Automation Conference (ASP-DAC), 2011. 16th Asia and South Pacific*, pp. 160–164, Jan. 2011.

[62] "ARM (2011), AMBA advanced extensible interface (AXI) protocol specification, version 2.0." http:www.arm.com.

[63] "OCP International Partnership, Open Core Protocol Specification. 2.0 Release Candidate, 2003."

[64] T. Bjerregaard, S. Mahadevan, R. Olsen, and J. Sparsoe, "An OCP compliant network adapter for GALS-based SoC design using the MANGO network-on-chip," in *2005 International Symposium on System-on-Chip, 2005. Proceedings*, pp. 171–174, Nov. 2005.

[65] "Philips Semiconductors, Device Transaction Level (DTL) Protocol Specification. Version 2.2, Jul. 2002."

[66] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Computing Surveys (CSUR)*, vol. 38, no. 1, p. 1, 2006.

[67] W. Jang and D. Pan, "An sdram-aware router for networks-on-chip," in *46th ACM/IEEE Design Automation Conference, 2009. DAC '09*, pp. 800–805, Jul. 2009.

[68] D. E. Culler, A. Gupta, and J. P. Singh, *Parallel Computer Architecture: A Hardware/Software Approach*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997.

[69] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, and H. Tenhunen, "A high-performance network interface architecture for NoCs using reorder buffer sharing," in *18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2010*, pp. 546–550, Feb. 2010.

[70] S. Das, A. Fan, K.-N. Chen, *et al.*, "Technology, performance, and computer-aided design of three-dimensional integrated circuits," in *Proceedings of the 2004 International Symposium on Physical Design*, ser. ISPD '04. New York, NY, USA: ACM, pp. 108–115, 2004.

[71] W. Davis, J. Wilson, S. Mick, *et al.*, "Demystifying 3D ICs: the pros and cons of going vertical," *Design Test of Computers, IEEE*, vol. 22, no. 6, pp. 498–510, Nov. 2005.

[72] C. Liu, I. Ganusov, M. Burtscher, and S. Tiwari, "Bridging the processor-memory performance gap with 3D IC technology," *Design Test of Computers, IEEE*, vol. 22, no. 6, pp. 556–564, Nov. 2005.

[73] J. Joyner, P. Zarkesh-Ha, and J. Meindl, "A stochastic global net-length distribution for a three-dimensional system-on-a-chip (3d-soc)," in *14th*

Annual IEEE International ASIC/SOC Conference, 2001. Proceedings, pp. 147–151, 2001.

[74] N. Madan and R. Balasubramonian, "Leveraging 3d technology for improved reliability," in *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 40. Washington, DC, USA: IEEE Computer Society, pp. 223–235, 2007.

[75] B. Goplen and S. Sapatnekar, "Efficient thermal placement of standard cells in 3D ICs using a force directed approach," in *International Conference on Computer Aided Design, 2003. ICCAD-2003*, pp. 86–89, Nov. 2003.

[76] J. Cong and Y. Zhang, "Thermal via planning for 3-D ICs," in *IEEE/ACM International Conference on Computer-Aided Design, 2005. ICCAD-2005*, pp. 745–752, Nov. 2005.

[77] B. Dang, P. Joseph, M. Bakir, *et al.*, "Wafer-level microfluidic cooling interconnects for gsi," in *Proceedings of the IEEE 2005 International Interconnect Technology Conference, 2005*, pp. 180–182, Jun. 2005.

[78] F. Catthoor, D. Verkest, and E. Brockmeyer, "Proposal for unified system design meta flow in task-level and instruction-level design technology research for multi-media applications," in *Proceedings. 11th International Symposium on System Synthesis, 1998*, pp. 89–95, Dec. 1998.

[79] M. Balakrishnan, A. Majumdar, D. Banerji, J. Linders, and J. Majithia, "Allocation of multiport memories in data path synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, no. 4, pp. 536–540, Apr. 1988.

[80] T. Kim and C. Liu, "Utilization of multiport memories in data path synthesis," in *30th Conference on Design Automation, 1993*, pp. 298–302, Jun. 1993.

[81] F. Balasa, F. Catthoor, and H. De Man, "Dataflow-driven memory allocation for multi-dimensional signal processing systems," in *IEEE/ACM International Conference on Computer-Aided Design, 1994*, pp. 31–34, Nov. 1994.

[82] R. Terrill and G. Beene, "3d packaging technology overview and mass memory applications," in *Aerospace Applications Conference, 1996. Proceedings, 1996 IEEE*, vol. 2, pp. 347–355, Feb. 1996.

[83] M. Monchiero, G. Palermo, C. Silvano, and O. Villa, "Exploration of distributed shared memory architectures for NoC-based multiprocessors," in *International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation, 2006. IC-SAMOS 2006*, pp. 144–151, Jul. 2006.

[84] A. Weldezion, Z. Lu, R. Weerasekera, and H. Tenhunen, "3-D memory organization and performance analysis for multi-processor network-on-chip architecture," in *IEEE International Conference on 3D System Integration, 2009. 3DIC 2009*, pp. 1–7, Sept. 2009.

[85] B. Goplen and S. Sapatnekar, "Efficient thermal placement of standard cells in 3D ICs using a force directed approach," in *International Conference on Computer Aided Design, 2003. ICCAD-2003*, pp. 86–89, Nov. 2003.

[86] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," in *IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004*. IEEE, pp. 306–313, 2004.

[87] J. Cong and Y. Zhang, "Thermal via planning for 3-D ICs," in *IEEE/ACM International Conference on Computer-Aided Design, 2005. ICCAD-2005*, pp. 745–752, Nov. 2005.

[88] B. Dang, P. Joseph, M. Bakir, *et al.*, "Wafer-level microfluidic cooling interconnects for GSI," in *Interconnect Technology Conference, 2005. Proceedings of the IEEE 2005 International*, pp. 180–182, Jun. 2005.

[89] S. Das, "Design automation and analysis of three-dimensional integrated circuits," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.

[90] A. W. Fang, H. Park, O. Cohen, *et al.*, "Electrically pumped hybrid algainas-silicon evanescent laser," *Optics Express*, vol. 14, no. 20, pp. 9203–9210, Oct. 2006.

[91] R. E. Camacho-Aguilera, Y. Cai, N. Patel, *et al.*, "An electrically pumped germanium laser," *Optics Express*, vol. 20, pp. 11 316–11 320, 2012.

[92] L. Liu, T. Spuesens, G. Roelkens, *et al.*, "A thermally tunable III–V compound semiconductor microdisk laser integrated on silicon-on-insulator circuits," *Photonics Technology Letters, IEEE*, vol. 22, no. 17, pp. 1270–1272, Sept. 2010.

[93] T. Wang, H. Liu, A. Lee, F. Pozzi, and A. Seeds, "1.3-μm InAs/GaAs quantum-dot lasers monolithically grown on Si substrates," *Optics Express*, vol. 19, no. 12, pp. 11 381–11 386, Jun. 2011.

[94] B. Moss, "High-speed modulation of resonant cmos photonic modulators in deep-submicron bulk-CMOS," Ph.D. dissertation, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 2009.

[95] L. Tong, R. R. Gattass, J. B. Ashcom, *et al.*, "Subwavelength-diameter silica wires for low-loss optical wave guiding," *Nature*, vol. 426, no. 0028-0836, pp. 816–819, Dec. 2003.

[96] J. Pang, C. Dwyer, and A. R. Lebeck, "More is less, less is more: molecular-scale photonic NoC power topologies," in *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '15, pp. 283–296, 2015.

[97] A. Shacham, B. Lee, A. Biberman, K. Bergman, and L. Carloni, "Photonic NoC for dma communications in chip multiprocessors," in *15th Annual IEEE Symposium on High-Performance Interconnects, 2007. (HOTI 2007)*, pp. 29–38, Aug. 2007.

[98] L. Zhang, X. Tan, M. Yang, *et al.*, "Circuit-switched on-chip photonic interconnection network," in *IEEE 9th International Conference on Group IV Photonics (GFP 2012)*, pp. 282–284, Aug. 2012.

[99] Y. Ye, J. Xu, X. Wu, *et al.*, "A torus-based hierarchical optical-electronic network-on-chip for multiprocessor system-on-chip," *ACM Journal on Emerging. Technologies in Computing*, vol. 8, no. 1, pp. 5:1–5:26, Feb. 2012. [Online]. Available: http://doi.acm.org/10.1145/2093145.2093150

[100] S. Koohi, Y. Yin, S. Hessabi, and S. J. B. Yoo, "Towards a scalable, low-power all-optical architecture for networks-on-chip," *ACM Trans. Embed. Comput. Syst.*, vol. 13, no. 3s, pp. 101:1–101:30, Mar. 2014.

[101] M. Bahmani, M. Reshadi, A. Khademzadeh, and A. Reza, "Corona: Ring-based interconnected topology for on-chip network," in *3rd International Design and Test Workshop, 2008 (IDT 2008)*, pp. 199–204, 2008.

[102] Y. Pan, P. Kumar, J. Kim, *et al.*, "Firefly: Illuminating future network-on-chip with nanophotonics," in *Proceedings of the 36th Annual International Symposium on Computer Architecture, 2009 (ISCA 2009)*, ser. ISCA '09, pp. 429–440, 2009.

[103] A. Joshi, C. Batten, Y.-J. Kwon, *et al.*, "Silicon-photonic clos networks for global on-chip communication," in *3rd ACM/IEEE International Symposium on Networks-on-Chip, 2009 (NoCS 2009)*, pp. 124–133, May 2009.

[104] C. Li, M. Browning, P. Gratz, and S. Palermo, "Luminoc: A power-efficient, high-performance, photonic network-on-chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 6, pp. 826–838, Jun. 2014.

[105] G. Kurian, J. E. Miller, J. Psota, *et al.*, "Atac: a 1000-core cache-coherent processor with on-chip optical network," in *Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques, 2010 (PACT 2010)*, ser. PACT '10, pp. 477–488, 2010.

[106] P. Grani and S. Bartolini, "Design options for optical ring interconnect in future client devices," *ACM Journal on Emerging. Technologies in Computing*, vol. 10, no. 4, pp. 30:1–30:25, Jun. 2014.

[107] S. Pasricha and S. Bahirat, "Opal: a multi-layer hybrid photonic NoC for 3D ICs," in *16th Asia and South Pacific Design Automation Conference, 2011 (ASP-DAC 2011)*, pp. 345–350, Jan. 2011.

[108] B. Ciftcioglu, J. Gao, R. Berman, *et al.*, "Recent progress on 3-d integrated intra-chip free-space optical interconnect," in *2012 IEEE Optical Interconnects Conference (OIC 2012)*, pp. 56–57, 2012.

[109] A. Abousamra, R. Melhem, and A. Jones, "Two-hop free-space based optical interconnects for chip multiprocessors," in *Fifth IEEE/ACM International Symposium on Networks on Chip, 2011 (NoCS 2011)*, pp. 89–96, May 2011.

[110] S. Bahirat and S. Pasricha, "A particle swarm optimization approach for synthesizing application-specific hybrid photonic networks-on-chip," in *13th International Symposium on Quality Electronic Design, 2012 (ISQED 2012)*, pp. 78–83, Mar. 2012.

[111] S. Woo, M. Ohara, E. Torrie, J. Singh, and A. Gupta, "The splash-2 programs: characterization and methodological considerations," in *Proceedings of the 22nd Annual International Symposium on Computer Architecture, 1995 (ISCA '95)*, pp. 24–36, Jun. 1995.

[112] M. Nikdast, J. Xu, X. Wu, *et al.*, "Systematic analysis of crosstalk noise in folded-torus-based optical networks-on-chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 3, pp. 437–450, Mar. 2014.

[113] J. Wang, J. Beu, R. Bheda, *et al.*, "Manifold: a parallel simulation framework for multicore systems," in *2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Mar. 2014.

[114] J. Miller, H. Kasture, G. Kurian, *et al.*, "Graphite: a distributed parallel simulator for multicores," in *IEEE 16th International Symposium on High Performance Computer Architecture (HPCA 2010)*, pp. 1–12, Jan. 2010.

[115] L. Carloni, P. Pande, and Y. Xie, "Networks-on-chip in emerging interconnect paradigms: advantages and challenges," in *3rd ACM/IEEE International Symposium on Networks-on-Chip, 2009. NoCS 2009*, pp. 93–102, May 2009.

[116] H. Matsutani, M. Koibuchi, I. Fujiwara, *et al.*, "Low-latency wireless 3d NoCs via randomized shortcut chips," in *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pp. 1–6, Mar. 2014.

[117] M. Schroeder, A. Birrell, M. Burrows, *et al.*, "Autonet: a high-speed, self-configuring local area network using point-to-point links," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1318–1335, Oct. 1991.

[118] A. Ganguly, K. Chang, S. Deb, *et al.*, "Scalable hybrid wireless network-on-chip architectures for multicore systems," *Computers, IEEE Transactions on*, vol. 60, no. 10, pp. 1485–1502, Oct. 2011.

[119] S. Hollis, C. Jackson, P. Bogdan, and R. Marculescu, "Exploiting emergence in on-chip interconnects," *IEEE Transactions on Computers*, vol. 63, no. 3, pp. 570–582, Mar. 2014.

[120] W. Davis, J. Wilson, S. Mick, *et al.*, "Demystifying 3D ICs: the pros and cons of going vertical," *Design Test of Computers, IEEE*, vol. 22, no. 6, pp. 498–510, Nov. 2005.

[121] S. Saito, Y. Kohama, Y. Sugimori, *et al.*, "Muccra-cube: A 3d dynamically reconfigurable processor with inductive-coupling link," in *International Conference on Field Programmable Logic and Applications, 2009. FPL 2009*, pp. 6–11, Aug. 2009.

[122] H. Matsutani, Y. Take, D. Sasaki, *et al.*, "A vertical bubble flow network using inductive-coupling for 3-d cmps," in *2011 Fifth IEEE/ACM International Symposium on Networks on Chip (NoCS)*, pp. 49–56, May 2011.

[123] H. Matsutani, P. Bogdan, R. Marculescu, *et al.*, "A case for wireless 3D NoCs for CMPs," in *Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific*, pp. 23–28, Jan. 2013.

[124] Z. Zhang, S. Yin, L. Liu, and S. Wei, "An inductive-coupling interconnected application-specific 3d NoC design," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 550–553, May 2013.

[125] A. Fazzi, L. Magagni, M. Mirandola, *et al.*, "3-d capacitive interconnections for wafer-level and die-level assembly," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 10, pp. 2270–2282, Oct. 2007.

[126] N. Miura, M. Inoue, K. Niitsu, *et al.*, "A 1tb/s 3w inductive-coupling transceiver for inter-chip clock and data link," in *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, pp. 1676–1685, Feb. 2006.

[127] A. More and B. Taskin, "Simulation based study of on-chip antennas for a reconfigurable hybrid 3d wireless NoC," in *SOC Conference (SOCC), 2010 IEEE International*, pp. 447–452, Sept. 2010.

[128] A. More and B. Taskin, "Simulation based study of on-chip antennas for a reconfigurable hybrid 3d wireless NoC," in *SOC Conference (SOCC), 2010 IEEE International*. IEEE, pp. 447–452, 2010.
[129] S.-B. Lee, S.-W. Tam, I. Pefkianakis, *et al.*, "A scalable micro wireless interconnect structure for cmps," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, pp. 217–228, 2009.
[130] T. Kikkawa, K. Kimoto, and S. Watanabe, "Ultrawideband characteristics of fractal dipole antennas integrated on Si for ULSI wireless interconnects," *Electron Device Letters, IEEE*, vol. 26, no. 10, pp. 767–769, Oct. 2005.
[131] IEEE, "Ieee standard for systemverilog – unified hardware design, specification, and verification language," *IEEE STD 1800-2009*, pp. 1–1285, Dec. 2009.
[132] "Systemc website." [Online]. Available: www.accellera.org
[133] "Omnet++ website." [Online]. Available: www.omnetpp.org
[134] "Synopsys platform architect." [Online]. Available: www.synopsys.com/Prototyping/ArchitectureDesign/Pages/platform-architect.aspx
[135] "Design player website." [Online]. Available: www.edautils.com/index.html
[136] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003.
[137] J.-Y. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, ser. Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg, vol. 2050, 2001.
[138] W. Fischer and K. Meier-Hellstern, "The Markov-modulated poisson process (mmpp) cookbook," *Performance Evaluation*, vol. 18, no. 2, pp. 149–171, 1993. [Online]. Available: www.sciencedirect.com/science/article/pii/016653169390035S
[139] "Orion 3 website." [Online]. Available: www.vlsicad.ucsd.edu/ORION3/index.html
[140] C. Sun, C.-H. Chen, G. Kurian, *et al.*, "Dsent – a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling," in *Sixth IEEE/ACM International Symposium on Networks on Chip (NoCS 2012)*, pp. 201–210, May 2012.
[141] "gem5 website." [Online]. Available: www.gem5.org
[142] "windriver simics website." [Online]. Available: www.windriver.com/products/simics
[143] "Sniper simulator website." [Online]. Available: www.snipersim.org
[144] "simplescalar simulator website." [Online]. Available: www.simplescalar.com
[145] K. Aisopos, C.-H. O. Chen, and L.-S. Peh, "Enabling system-level modeling of variation-induced faults in networks-on-chip," in *Proceedings of the 48th Design Automation Conference*, ser. DAC 2011, 2011. [Online]. Available: www.princeton.edu/∼carch/kaisopos/FaultModel
[146] "hotspot website." [Online]. Available: www.lava.cs.virginia.edu/HotSpot
[147] "3d-ice tools." [Online]. Available: www.esl.epfl.ch/3D-ICE

[148] P. Pande, H. Zhu, A. Ganguly, and C. Grecu, "Energy reduction through crosstalk avoidance coding in NoC paradigm," in *9th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006*, pp. 689–695, 2006.

[149] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*, ser. Prentice-Hall computer applications in electrical engineering series. Prentice-Hall, India, 1983.

[150] W. Dally, "Virtual-channel flow control," *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, no. 2, pp. 194–205, Mar. 1992.

[151] Y. Tamir and G. L. Frazier, "High-performance multi-queue buffers for vlsi communications switches," in *Proceedings of the 15th Annual International Symposium on Computer Architecture*, ser. ISCA '88. Los Alamitos, CA, USA: IEEE Computer Society Press, pp. 343–354, 1988. [Online]. Available: http://dl.acm.org/citation.cfm?id=52400.52439

[152] T. Lehtonen, P. Liljeberg, and J. Plosila, "Online reconfigurable self-timed links for fault tolerant NoC," *VLSI Design*, vol. 2007, 2007.

[153] H. Zimmer and A. Jantsch, "A fault model notation and error-control scheme for switch-to-switch buses in a network-on-chip," in *Hardware/Software Codesign and System Synthesis, 2003. First IEEE/ACM/IFIP International Conference on*, pp. 188–193, Oct. 2003.

[154] D. Rossi, P. Angelini, and C. Metra, "Configurable error control scheme for NoC signal integrity," in *On-Line Testing Symposium, 2007. IOLTS 07. 13th IEEE International*, pp. 43–48, Jul. 2007.

[155] M. Notomi, A. Shinya, K. Nozaki, *et al.*, "Low-power nanophotonic devices based on photonic crystals towards dense photonic network on chip," *Circuits, Devices Systems, IET*, vol. 5, no. 2, pp. 84–93, Mar. 2011.

Chapter 12

A new paradigm towards performance centric computation beyond CMOS: DNA computing

Prasun Ghosal[1], *Mayukh Sarkar*[1] *and Pratima Chatterjee*[1]

With rapid growth in very-large-scale integration technology following Moore's law, the integration density of transistors has reached billions. This caused scaling of transistors to reach deep submicron regime resulting in failure of classical physics. Eventually, classical computing technologies have reached a physical limit and caused slowing down of Moore's law. Also, current leakage becomes a major problem in classical technology at such small size that heats up the chip. So Dennard scaling, which states about the constant power density with the decrease of transistor size also failed. It caused the switch to multi-core technology but that too seems to be at the end due to *Dark Silicon* issues. As a plausible alternative, researchers are trying to switch to some non-Complementary Metal-Oxide Semiconductor (CMOS) technology, e.g., quantum computing, bio-inspired computing such as deoxyribonucleic acid (DNA), etc. Besides mitigating the concerns faced in conventional technology, DNA computing comes with a bunch of other benefits too to cater the needs of future-generations computing, viz. massively parallel operations, huge information density over silicon, etc. In this chapter, with an introduction to structure of DNA and how DNA computing works, several fields of DNA computing have been explored followed by how DNA computing can be applied to solve several problems otherwise known as *hard* on conventional computer with some comments on possible future research directions in this promising field.

12.1 Introduction

History of genetics starts with the laws of inheritance discovered by Gregor Mendel, an Austrian monk [1]. While performing hybridization experiments with several pea plants between 1856 and 1863, he discovered that a plant inherits one trait from each parent at each generation and the trait from one parent remains dominant over the same

[1]Indian Institute of Engineering Science and Technology, Shibpur, India

from other parent. Also, genes for different traits assort independently of one another. On the other hand, Johannes Friedrich Miescher, a Swiss physician, while working at the University of Tübingen, Germany, in 1869, isolated various phosphate-rich cellular material that was not protein and named it nuclein (now nucleic acids) [2]. The discovery of genetics was finalized by rediscovery of Mendel's experiment in 1900 by Hugo de Vries, Erich von Tschermak, and Carl Correns [4, 3, 5]. In 1928, Frederick Griffith discovered the *transforming principle* while studying the effect of injecting virulent smooth strain bacteria and non-virulent rough strain bacteria in mice [6]. The principle states that information can be transferred between different strains of bacteria. In 1943, Oswald Theodore Avery discovered [7] that the deoxyribonucleic acid (DNA) was transforming agent in Griffith's experiment [8]. Before that, in 1935, DNA has already been extracted in pure state for the first time by Russian Scientist Andrey Nikolayevich Belozersky [9]. In 1953, Watson and Crick discovered the double helical structure of DNA molecule [11, 10].

DNA molecule is generally present in the nucleus of each living cell and uniquely identifies the characteristics of that living organism. Beside nuclear DNA some complex organisms, e.g., human have small amount of DNA present in cell mitochondria that is also referred to as mitochondrial DNA (mtDNA).

12.1.1 DNA structure

DNA is a polymer nucleic acid formed by two strands coiled around each other to form double helical structure. Each of these two strands is composed of monomers known as nucleotides. Each nucleotide is composed of a nucleoside and a phosphate group. A nucleoside, in turn, is composed of a 5-carbon (pentose) monosaccharide sugar (deoxyribose) and a nitrogen containing nucleobase is attached to the sugar. These sugar and phosphate groups form DNA backbone that is a chain of alternating sugar and phosphate groups. Here, two adjacent sugar molecules are connected via a phosphate group that forms phosphodiester bond between the 3′- and 5′-carbon of the adjacent sugars. This deoxyribose sugar lacks an hydroxyl group at 2′-position, as opposed to the ribose sugar found in RNA. The base is attached to the sugar at the 1′-carbon. So each DNA strand has a direction with a 3′-end and a 5′-end where a terminal hydroxyl group is attached to the 3′-end and a phosphate group attached to the 5′-end.

Four types of nucleotide bases are found in a DNA molecule, viz. adenine (A), guanine (G), thiamine (T), and cytosine (C). Among these four bases, cytosine and thiamine are pyrimidine bases, i.e., they contain one six-member carbon–nitrogen ring and two nitrogen atoms at 1 and 3 positions of the ring. Adenine and guanine are purine bases with a pyrimidine ring fused to an imidazole ring containing two carbon–nitrogen rings and four nitrogen atoms. The chemical structures of the nucleotide bases are shown in Figure 12.1.

In a DNA double helix, two DNA strands are joined in anti-parallel fashion, i.e., run in opposite directions. Here, each base of one strand gets paired up with corresponding Watson–Crick complementary base of other strand by hydrogen bonds, i.e., adenine gets paired up with thiamine, and cytosine gets paired up with guanine.

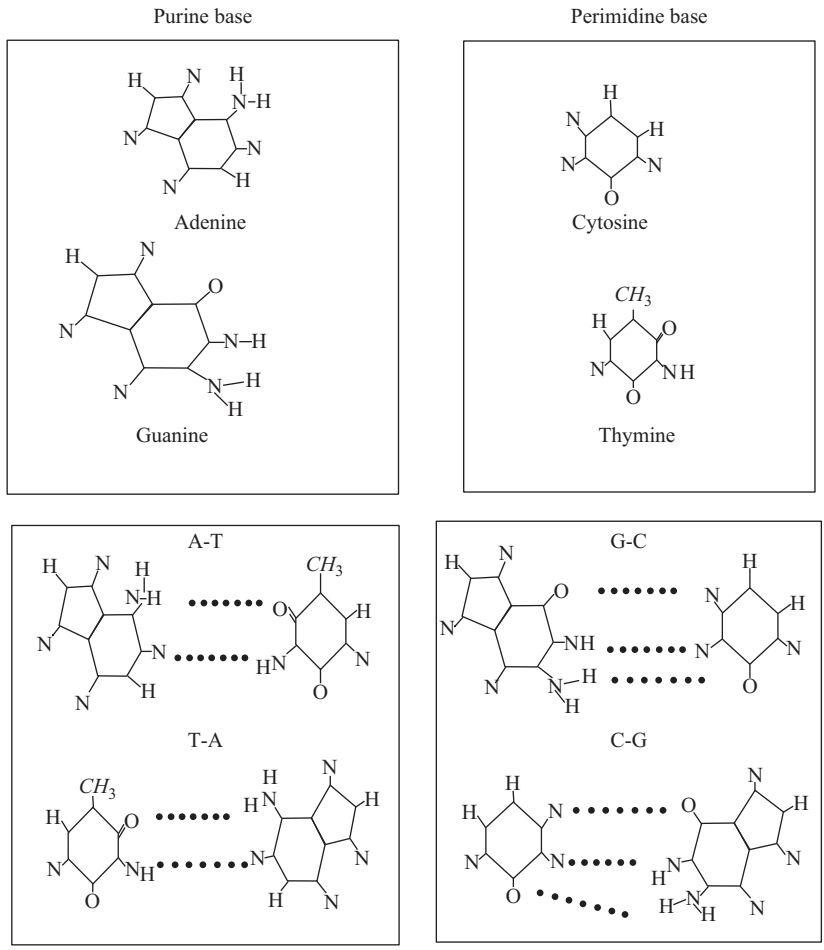


Figure 12.1 Chemical structures of the nucleotide bases

A-T pair has two hydrogen bonds and G-C pair has three hydrogen bonds. Structure of a DNA double helix is shown in Figure 12.2.

12.1.2 Operations on DNA solutions

The algorithms designed to solve computational problems using DNA are actually sequences of bio-molecular operations applicable on a solution of DNA molecules. Following operations are applicable on a solution of DNA computer.

Synthesis: DNA strands can be synthesized according to DNA sequences entered by user using a machine named "DNA Synthesizer". Four nucleotide bases are supplied to the synthesizer in a solution and the synthesizer creates millions of copies of desired strand.

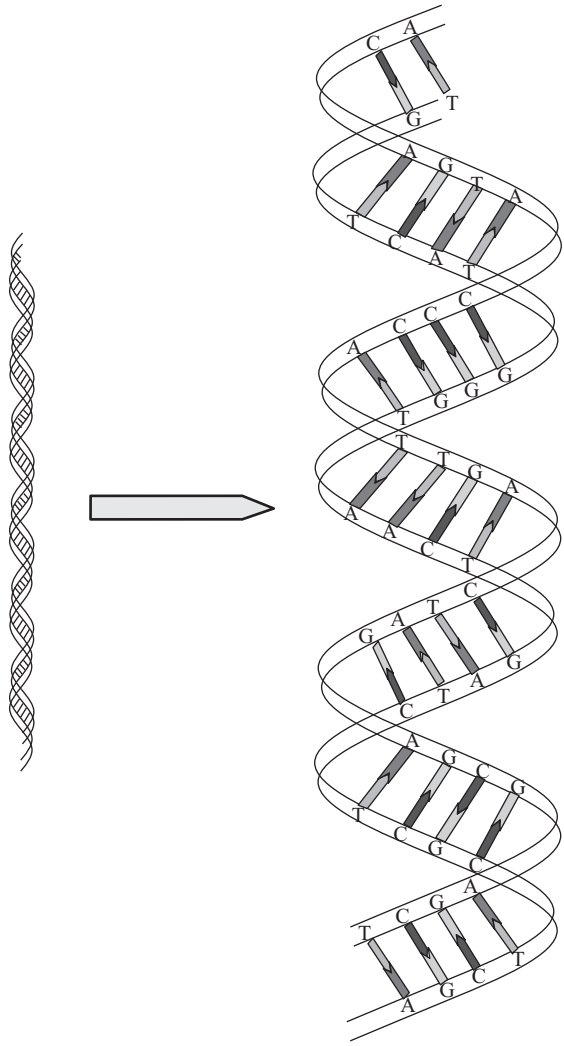


Figure 12.2 Structure of the DNA double helix

Denature: DNA double strands in a solution can be heated to get separated in two single strands as heat breaks the hydrogen bond between complementary pairs of double strand. As G-C pair has one more hydrogen bond than A-T pair, temperature required to break a G-C pair is also higher than that required for an A-T pair. So the denaturing temperature of a DNA strand primarily depends on nucleotide sequence. Strands having higher count of G-C pair also has higher denaturing temperature than other strands. Denature operation is shown in Figure 12.3.

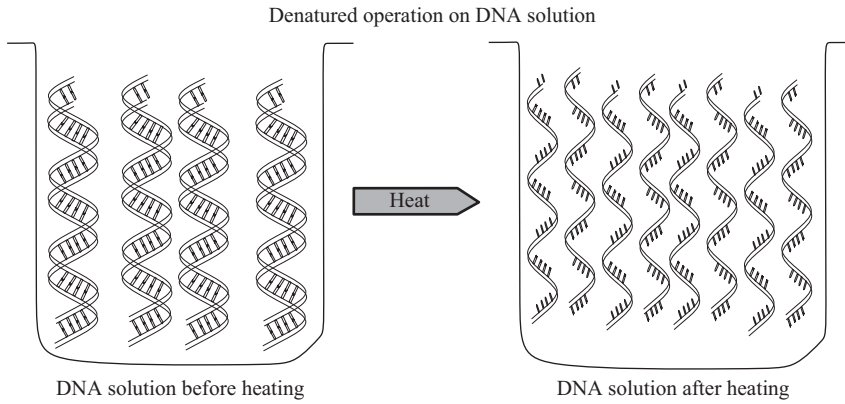


Figure 12.3 DNA denaturation

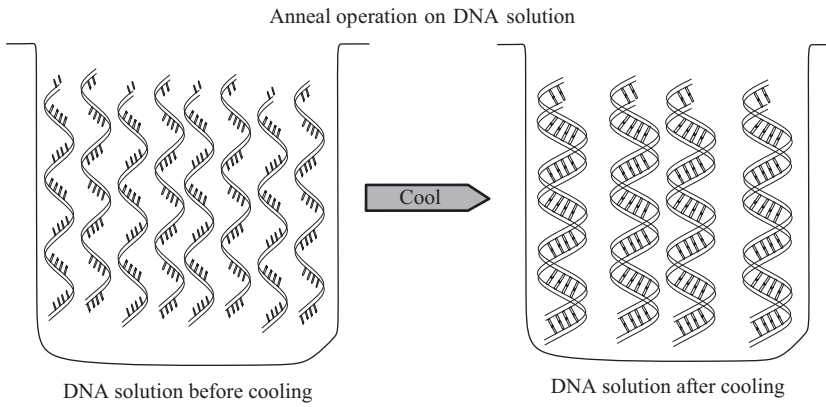


Figure 12.4 DNA anneal

Anneal: This operation is exactly opposite to denature and is also referred to as "hybridization". When a solution of single strand is cooled down then complementary strands get bound to each other and form double strand during random motion of DNA molecules. Anneal operation is shown in Figure 12.4.

Ligation: DNA ligase is a specific type of enzyme that joins two DNA strands by catalyzing the formation of covalent phosphodiester bond between the 3′-hydroxyl end of one nucleotide and 5′-phosphate end of other nucleotide. To perform ligation, adenosine triphosphate (ATP) is needed to be added along with ligase enzyme.

Amplify (PCR): Polymerase chain reaction (PCR) is used to make multiple copies of a DNA molecule. To perform this operation, solution containing double strands is first heated enough to get broken into single strands. Primers, complementary

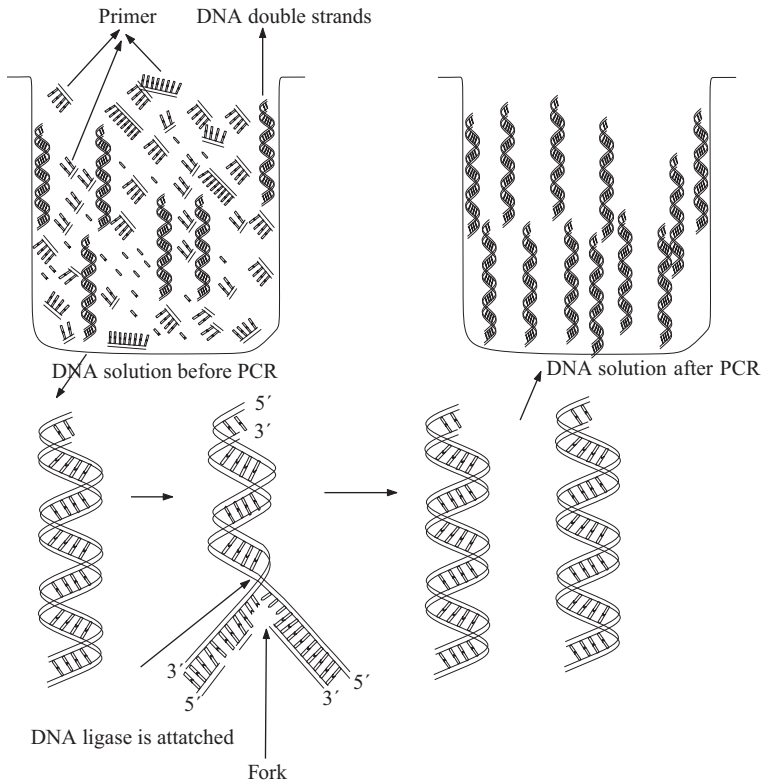


Figure 12.5 Polymerase chain reaction

strands of DNA molecule or part of molecule of interest, are added to solution along with *Taq* polymerase enzyme. This solution of broken single strands is cooled down to get primers attached and thus polymerase get bound and begin copying the DNA strand. This cycle is repeated several times to get millions of copies of the molecule. During each cycle number of DNA molecules get doubled in solution, i.e., the number of DNA molecules becomes 2^n after n number of steps. Steps of a PCR operation are shown in Figure 12.5.

Separation by string: DNA strands from a solution can be separated based on the presence of a particular short sequence. For example, let us consider separating DNA strands containing the sequence AGCAGTC. Magnetic bead separation technique can be used to perform this operation. Here, complementary oligos of the desired sequence are attached to tiny magnetic beads and added to the solution. Strands containing the target sequence get attached to the beads after annealing. A magnet is used to pull the beads out of the solution. An example of separation is shown in Figure 12.6 where the separation is being performed based on the presence of a substring ACGC. So the magnetic beads containing TGCG are being added to the solution.

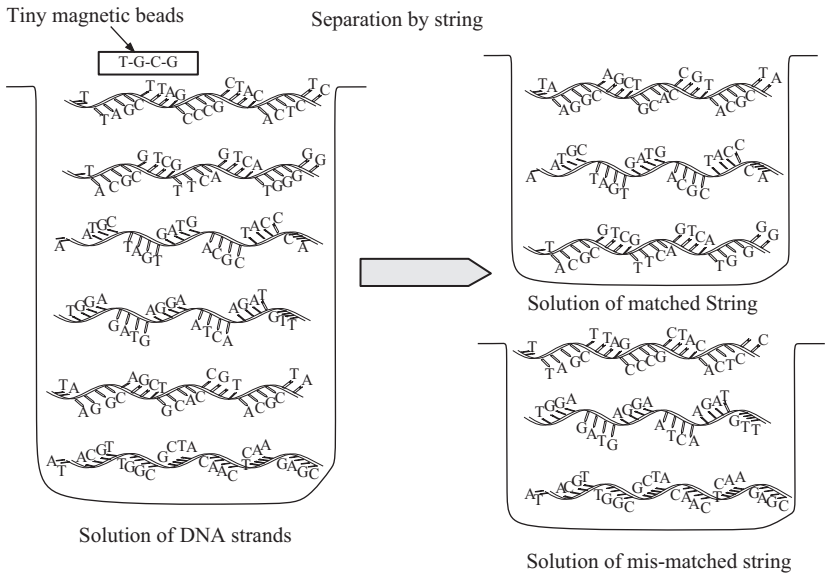


Figure 12.6 Separating DNA strands based on the presence of substring

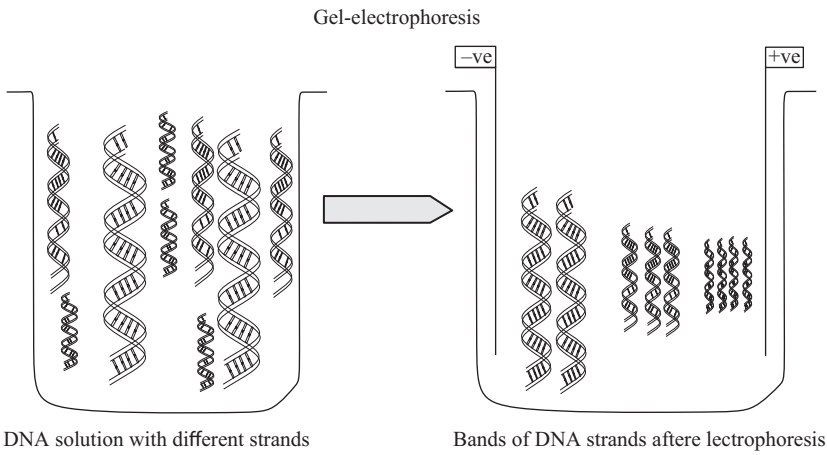


Figure 12.7 Separating DNA strands based on length

Separation by length: When DNA molecules are placed in an electric field, they get attracted towards the positive pole being negatively charged. When this electrophoresis is performed in a gel (agarose, polyacrylamide), strands move at different speeds based on their size. Smaller molecules move faster while larger ones move slower. This technique is called Gel Electrophoresis as shown in Figure 12.7. The result may be visualized by staining it using ethidium bromide and viewing under ultraviolet light.

12.1.3 How DNA computers work? Power of DNA computer

Problems are solved on a DNA computer by encoding problems using A, T, G, and C, synthesizing corresponding DNA strands, and performing suitable sequence of bio-molecular operations on those strands in a molecular biological laboratory. The finite sequence of bio-molecular operations used to solve the particular problem may be referred to as *DNA algorithm* for that problem. Output of DNA algorithms is DNA molecules too. They need to be decoded to get desired output in terms of main problem definition.

The main power of DNA computing over conventional ones is as follows:

1. **Massively parallel operation:** A single test tube of DNA can contain trillions of DNA strands and all strands respond to the biological operations in parallel.
2. **Huge information density of DNA over silicon:** Estimated storage capacity of 2.2 petabytes per gram of DNA has been reported in Reference 12.

Advantage of huge information density has proven DNA to be an important and reliable media to store information. Recent studies have also revealed that data stored on DNA molecule can last even thousands of years [13]. It also has attracted many researchers for being a primary application of DNA molecules as long term and reliable data storage [12, 14, 15].

On the other hand, among several other applications, designing strong cryptography and data hiding techniques have been invented over the years. In fact, even simplest cryptography technique using DNA molecules with original data encoded by DNA strand is mixed in a solution with millions of other dummy molecules and two 20-base primers are used as key proves to be more powerful than 56-bit Data Encryption Standard (DES) cryptography technique. This is due to the fact that key space for DNA method (4^{40}) is much larger than key space of 56-bit DES (2^{56}). Besides these hard problem of finding exact DNA molecule without primers, another hard biological problem, viz. *Fragment Assembly Problem* has also been used to encrypt messages [16, 17]. This power of hiding messages by DNA molecules has been proven from time to time by various researchers [18–23]. This power of encrypting messages has also been used in encrypting images using DNA molecules [24, 25].

12.1.4 History of DNA computing

12.1.4.1 Emergence for hard problem solving

In 1994, Adleman [26] has shown a new way of solving Nondeterministic Polynomial time (NP)-complete problems using DNA by solving Traveling Salesman Problem. Since then, many scientists have chosen this powerful tool to solve various NP-complete problems. In the very next year (1995), Lipton [27] developed an algorithm to solve the Boolean satisfiability problem (SAT) on a DNA computer in linear time. In the same year (1995), Boneh et al. [28] broke DES, a famous encryption method, using DNA. In 1997, Ouyang et al. [29] solved maximal clique problem given a six-vertex graph. A huge achievement was made by Adleman and others in 2002 when

they solved 20-variable 3-SAT problem by performing an exhaustive search over one million possibilities [30]. Along with these experiments, several other NP-complete problems, e.g., Graph Coloring [31], Bin Packing [32], etc., have been solved using DNA computing.

12.1.4.2 Attempts made for physical realization

In 2002, researchers from Weizmann Institute of Science, Israel have developed a programmable molecular computing machine composed of enzymes and DNA molecules. In 2003, the same team advanced one step further. In the new device, the single DNA molecule that provides the computer with input data also provides all necessary fuel [33]. In 2004, Benenson et al. [34] described an autonomous bio-molecular computer that logically analyzes the levels of messenger RNA species. In response, it produces a molecule capable of affecting levels of gene expression. This computer would be capable of diagnosing cancer theoretically and producing anti-cancer drug. In 2013, Goldman et al. [12] encoded computer files totaling 739 kB of hard disk storage and with an estimated Shannon information of 5.2×10^6 bits into a DNA code, synthesized, sequenced, and then reconstructed the original data with 100% accuracy. In the same year (2013), bio-engineers at Stanford University created first biological transistors named *transcriptor* using DNA and RNA [35]. On 26 October 2014, Israeli scientists in collaboration with researchers from around the world have developed DNA strands capable of carrying electrical charges for DNA-based electrical circuits [36, 37]. They have reported reproducible charge transport in guanine-quadruplex (G4) DNA molecules adsorbed on a mica substrate and have measured currents of tens of picoamperes to more than 100 pA in G4-DNA over distances ranging from tens of nanometers to more than 100 nm.

12.2 DNA computing models

Several abstract models of computations using DNA molecules have been proposed in this section. Each of these models is characterized by the set of operations available in these models. Some basic operations, e.g., merging two test tubes or copying contents of one test tube into other are available in all the models. Rest of the operations uniquely characterize the model. Algorithms designed for a DNA computer are sequenced in these operations and the approach to solve a particular problem may vary from one model to another. But these models are very much restricted by the success of implementations of these operations in bio-molecular laboratory. Several things are needed to keep in mind to understand the real difference between these abstract models and practical implementations of operations comprising the model as follows:

1. Time required to perform the operations in laboratory may vary vastly from one operation to another. Thus, the original time required to perform one sequence of operations may vary largely from another sequence of operations having same number of operations. But for the ease of analysis, the complexity of an algorithm is measured in terms of number of operations in the algorithm.

2. These laboratory techniques are not free from errors. So, the success of practical implementations of the algorithms depend on how much error-prone the implementation of operations comprising the algorithm are. But several measures can be taken to reduce the effect of these errors in some cases. For example, some DNA molecules may be lost while pouring DNA solution from one test tube to another. To assure that no DNA code get lost, millions of copies of each DNA strand are kept in the solution.

In these models, the DNA solution in a test tube is considered to be a multi-set of strings, where each string is made up of four letters A, G, C, and T. But for the ease of representation, these strings may be represented by some symbols or group of symbols from an alphabet.

12.2.1 Adleman–Lipton model

The Adleman–Lipton model, also known as the filtering model, is the first model proposed for DNA computing. Adleman performed basic bio-molecular operations on DNA solution to determine whether a graph has Hamiltonian path. Lipton used the same technique to solve the SAT problem, a computationally "hard" problem for a conventional computer in reasonable time.

Idea of filtering model is to first generate all possible solutions to the problem and then filtering out the solutions violating the criteria. These possible solutions are represented using suitable DNA strands in a solution and then a sequence of bio-molecular operations is used to filter out the undesirable DNA strands. As for example, to solve the *Hamiltonian path problem*, Adleman generated all possible paths of the graph and then step by step removed those paths that cannot be Hamiltonian. Similarly, to solve the *three-vertex-colorability*, first all possible colorings of the graph are generated and then those colorings are removed that violate the criteria, i.e., have two adjacent vertices with same color.

The basic Adleman–Lipton model thus contains the following operations:

1. *separate*(T, S): Separate DNA strands of test tube T in two parts, one having S as substring and strands in other part do not contain S. Place two parts in two test tubes, T_S^{on} contains the strands having S as substring, and T_S^{off} contains the others.
2. *merge*$(T : T_1, T_2, \ldots, T_n)$: Pour the contents of test tubes $T_1, T_2, \ldots, T_n$ in test tube T and thus forming $T = \cup(T, T_1, T_2, \ldots, T_n)$.
3. *detect*(T): Given a test tube T, detect whether the test tube is empty or not. If empty, return *false*, otherwise return *true*.

12.2.1.1 Adleman's experiment: solving Hamiltonian path problem

In 1996, Adleman first showed that DNA molecules those were otherwise used to be considered only as the information to carry the characteristic of a living organism can also be used to solve computational problems. So, in a sense, this work is the starting of the DNA computation. Adleman solved the Hamiltonian path problem that states whether a graph has a Hamiltonian path given a starting vertex and an end vertex.

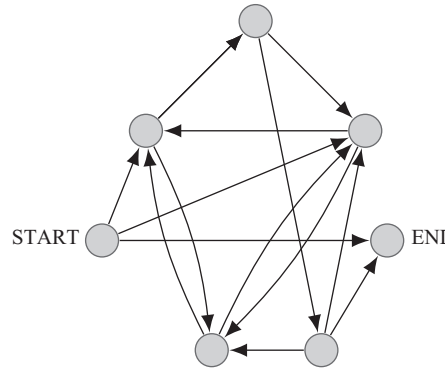


Figure 12.8 The graph used by Adleman in original experiment (Source: Reference [26])

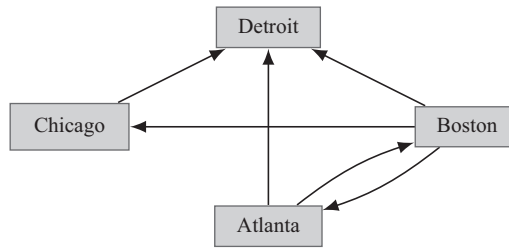


Figure 12.9 The graph used by Adleman for demonstration (Source: Reference [26])

The Hamiltonian path is a path in graph, which traverses through all the nodes of the graph exactly once. Adleman spent 7 days in lab to find the Hamiltonian path in a graph. Though this time seems long, but if we consider the large and uneven times taken by various experiments on a DNA solution and understand the fact that algorithm devised by Adleman is linear in terms of count of operations, whereas the problem itself is computationally "hard" for a conventional computer, we can readily understand the importance of the work. Although Adleman performed this operation on a seven-vertex graph with 14 edges as shown in Figure 12.8, but to understand the representation of the problem in terms of DNA code and the sequence of operations performed on the solution step by step, Adleman provided an example on a small prototype with a graph having four nodes and six edges as shown in Figure 12.9. We, for the sake of easy understanding, will use same prototype used by Adleman.

A bare-bone procedure to solve the Hamiltonian path problem may be viewed as follows:

1. Generate all possible paths in the graph.
2. For each path in the set, do the following:
   a. Check whether the path begins with the given start vertex and ends with the given end vertex. If not, discard the path.

b. Check whether the path passes through exactly *n* number of nodes, where *n* is the number of nodes in the graph. If not, discard the path.
c. For each vertex, check whether the path passes through that vertex. If not, discard the path.

3. Check whether the path set is empty. If not empty, report that there is a Hamiltonian path, otherwise, there is no Hamiltonian path in the graph.

As first step, elements of the problem, i.e., in this case, nodes and edges need to be properly encoded. Adleman assigned random DNA sequence to each node. For the prototype example, let us consider an assignment where each node is assigned 8-mer DNA sequence as shown in Table 12.1.

Now, each node may be considered to have a first half and a second half. For example, Boston may have the first half as TCGG and second half as ACTG. For edges, each edge is encoded with 8-mer DNA strand obtained by concatenating the second half of source node and first half of destination node. For example, Atlanta–Boston edge is encoded by concatenating the second half of Atlanta (GCAG) and the first half of Boston (TCGG) thus obtaining GCAGTCGG. In this manner, all six edges can be encoded as follows in Table 12.2.

Now let us assume that we need to find whether any Hamiltonian path exists that starts from Atlanta and ends to Detroit. As we can see, there exists a Hamiltonian path Atlanta–Boston–Chicago–Detroit. So the output of the algorithm should keep this path after filtering out the unnecessary paths. So the final solution must contain the DNA sequence GCAGTCGGACTGGGCTATGTCCGA, whereas the first 8-mer represents the edge Atlanta–Boston, second 8-mer as Boston–Chicago and the

Table 12.1 DNA sequence assigned to the nodes

Node	DNA sequence	Watson–Crick complement
Atlanta	ACTTGCAG	TGAACGTC
Boston	TCGGACTG	AGCCTGAC
Chicago	GGCTATGT	CCGATACA
Detroit	CCGAGCAA	GGCTCGTT

Table 12.2 DNA sequence assigned to the edges

Edge	DNA sequence
Atlanta–Boston	GCAGTCGG
Atlanta–Detroit	GCAGCCGA
Boston–Chicago	ACTGGGCT
Boston–Detroit	ACTGCCGA
Boston–Atlanta	ACTGACTT
Chicago–Detroit	ATGTCCGA

last 8-mer Chicago–Detroit, as clear from Table 12.2. Now the algorithm works as follows:

1. Synthesize the Watson–Crick complementary sequences for each node and sequence for each edge and put them in a single test tube. So, the test tube T contains following DNA strands.

   T = {TGAACGTC, AGCCTGAC, CCGATACA, GGCTCGTT, GCAGTCGG, GCAGCCGA, ACTGGGCT, ACTGCCGA, ACTGACTT, ATGTCCGA}

2. Next add ligase enzyme and other ingredients to facilitate the ligation operation in the test tube. During random movement of molecules in the solution it may happen that the complementary DNA strand encoding Boston (AGCCTGAC) and the strand encoding Atlanta–Boston edge (GCAGTCGG) may come closer and as the last half of Atlanta–Boston edge (TCGG) and the first half of Boston node (AGCC) are complementary they form double strand as

   $$\begin{pmatrix} \text{GCAGTCGG} \\ \text{AGCCTGAC} \end{pmatrix}$$

   Now the resulting complex has TGAC as the sticky end. If the Boston–Chicago edge comes closer to this complex, it will join to the sticky end and the new complex thus formed becomes

   $$\begin{pmatrix} \text{GCAGTCGGACTGGGCT} \\ \text{AGCCTGAC} \end{pmatrix}$$

   Similarly, the complementary node of Chicago (CCGATACA) may find the new sticky end of the complex (GGCT) and get attached. The ligase enzyme joins the edge strands by creating phosphodiester bond between them. In this manner, various random paths are formed. If suitably high number of molecules are taken at the beginning, then clearly there is a very high probability that most of the paths are generated and if not all they at least contain the Hamiltonian path.
3. As nearly all possible paths are found now, we need to filter out unnecessary ones. To begin that, first perform PCR with two primers, last half of the source node (in this case, as the starting node is Atlanta, one primer is GCAG) and the first half of the destination node (in this case, Detroit and hence, CCGA). All the DNA strands that begin with GCAG and end with CCGA are amplified at exponential rate. The strands that begin with GCAG but do not end with CCGA or vice versa are amplified at much slower rate, almost linearly. Those strands that neither start with GCAG nor end with CCGA are not amplified at all. Adleman took a small amount of the solution after PCR. Thus, there is a high probability that most of the strands having GCAG at the beginning and CCGA at the end were obtained along with very small amount of other molecules.
4. Now, we have those DNA strands that starts with Atlanta and ends with Detroit. Next, Gel Electrophoresis is performed and DNA strands are separated having the right length. (For a graph with n cities and each node and edge being

p-mer, separate the band containing $(n-1)p$-mer DNA strands, as the possible Hamiltonian paths will contain $(n-1)$ edges. In the example, separate the bands with 24-mer DNA strands.)

5. Now, we have the paths having correct start node and end node, and have $(n-1)$ edges, where n is the number of nodes. Now, the affinity purification (separation by string) needs to be performed for each intermediate node using magnetic bead separation method to assure it is going through all the nodes. In this case, Boston and Chicago are the intermediate nodes. First, use the complement of Boston node as the probes on beads and pull the strands having Boston (TCGGACTG) as substring. After separating the strands having Boston, use the filtered out strands to re-filter against the Chicago node with complement of Chicago as the probes on beads.
6. After performing the affinity purification for all intermediate nodes, we thus have all strands with correct start and end nodes having $(n-1)$ edges and all nodes are present and hence the Hamiltonian paths. So, if at end, test tube is empty, the graph has no Hamiltonian path with given starting and end nodes. Otherwise, the test tube contains the desired Hamiltonian path.

12.2.1.2 Lipton's experiment: solving SAT problem

A Boolean expression is made up of Boolean variables and Boolean operators AND (denoted by $\wedge$), OR (denoted by $\vee$), and NOT (or negation, denoted by $\neg$). An expression is called *satisfiable*, if there exists an assignments of variables that makes the expression TRUE. SAT is therefore, given a Boolean expression, determining whether the expression is satisfiable. The Boolean expressions are built up as conjunction (AND) of *clause*s, where each clause is disjunction (OR) of *literal*s, and each literal is either a variable (positive literal) or negation of the variable (negative literal). The SAT problem is one of the first proven NP-complete problems.

Lipton solved the SAT problem using linear complexity of bio-molecular operations. The idea was to represent the problem using graph and then solve the problem using Adleman's approach. Assume that the variable set of the problem is $V = \{x_1, x_2, \ldots, x_n\}$. The graph is $(3n+1)$-node graph having the node set $\{a_1, x_1, x_1', a_2, x_2, x_2', \ldots, a_n, x_n, x_n', a_{n+1}\}$, with edges from a_k to x_k, a_k to x_k', x_k to a_{k+1}, x_k' to a_{k+1} for $1 \leq k \leq n$, as shown in Figure 12.10 for two-variable set $\{x, y\}$.

Binary strings are represented by paths from a_1 to a_{n+1}, where the path passes through x_k if $x_k = 1$ in the assignment, and the path passes through x_k' if $x_k = 0$ in the assignment. The nodes and edges of the graph are encoded in the similar manner to the Adleman's representation. Also, the initial set of all possible assignments, i.e., all possible paths in the graph are also generated using Adleman's procedure.

Assume that the expression consists of m clauses, $C_1, C_2, \ldots, C_m$. We need to check whether there exists any assignment that makes each of these clauses TRUE. To obtain that, for each clause, one by one, the paths making the clause TRUE needs to be filtered and used for the next clause.

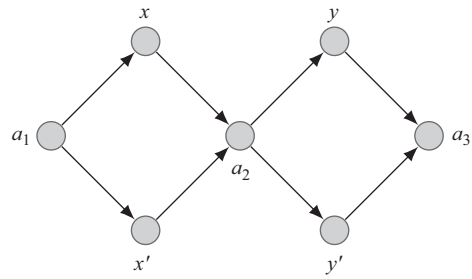


Figure 12.10 Lipton's SAT solver graph (Source: Reference [27])

Algorithm 12.1 thus works as follows:

Algorithm 12.1 Lipton's SAT solver.

```
1: procedure SATSolver(in : The initial path set in test tube T_0)
2:     for each clause C_i for 1 ≤ i ≤ m do
3:         for each literal v_j in C_i do
4:             if v_j is a positive literal then
5:                 T_i^j ← T_{i-1}^{on}_{v_j}
6:                                   ▷ Separate those paths from T_{i-1} which have v_j
                                      as substring
7:             else
8:                 T_i^j ← T_{i-1}^{on}_{v'_j}
9:                                   ▷ Separate those paths from T_{i-1} which have v'_j
                                      as substring
10:            end if
11:        end for
12:        Merge all T_i^j s in T_i
13:                                  ▷ T_i now contains those paths that satisfy
                                      C_1, C_2, ..., C_i
14:    end for
15:    Detect(T_n)                   ▷ If T_n is non-empty, the expression is satisfiable,
                                      otherwise not
16: end procedure
```

12.2.2 Sticker model

A binary number may be represented in the DNA sticker model by employing two groups of single-stranded DNA molecules. One is *memory strand*, which is a long DNA molecule, subdivided into several non-overlapping region. Other group is a set

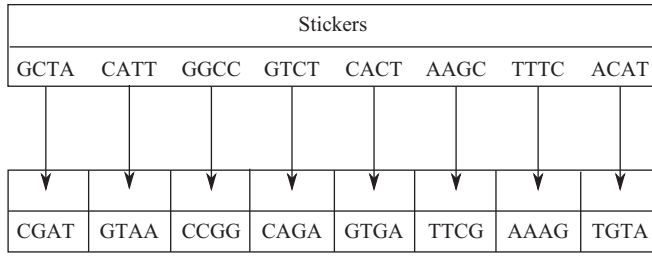


Figure 12.11 An example memory strand with corresponding stickers

GCTA		GGCC		CACT	AAGC	TTTC	
CGAT	GTAA	CCGG	CAGA	GTGA	TTCG	AAAG	TGTA
1	0	1	0	1	1	1	0

GCTA	CATT		GTCT	CACT			ACAT
CGAT	GTAA	CCGG	CAGA	GTGA	TTCG	AAAG	TGTA
1	1	0	1	1	0	0	1

Figure 12.12 An example memory strand with corresponding stickers

of *stickers*, which are short DNA molecules, each having length equal to length of each region of memory strand. Each sticker is complementary to one and only one of the non-overlapping regions. Each non-overlapping region represents a bit. If a sticker is annealed to its matching region on the memory strand that region then represents 1 bit, otherwise a 0 bit.

For example, to represent a 8-bit number, if we take the memory strand and the corresponding stickers as in Figure 12.11, where each sticker is complementary to each of the eight non-overlapping regions, respectively. As an example, the numbers 11011001 and 01001110 can be represented using the above sticker model as in Figure 12.12.

Any set of bit strings can be represented by identical memory strands, each memory strand having stickers annealed only at the required 1 bit positions.

12.2.2.1 Operations on sticker-based DNA

The operations available on sticker-based DNA strands are as follows:

1. ***Combine***: In this operation, two sets of bit strings in two test tubes are combined in one test tube. This corresponds to producing a new tube containing all the memory complexes from both input tubes.
2. ***Separate***: In this operation, a set of strings is separated into two sets based on a particular bit. This creates two new tubes, where one tube contains strings having that particular bit *on*, and the other tube contains the strings having the bit *off*.

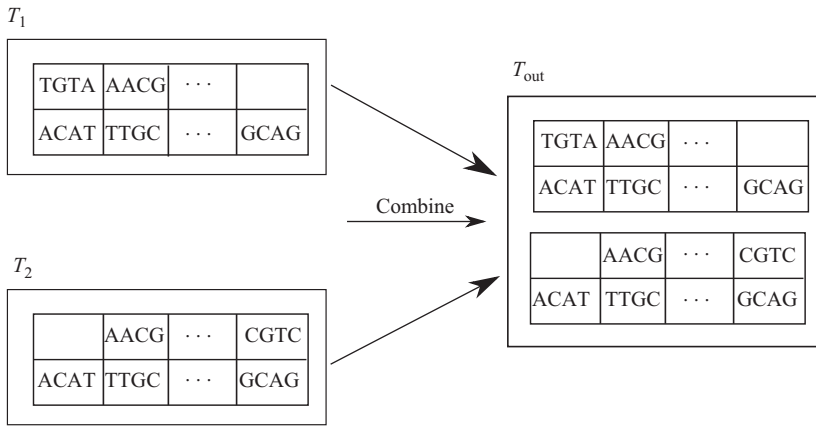


Figure 12.13 Combine operation

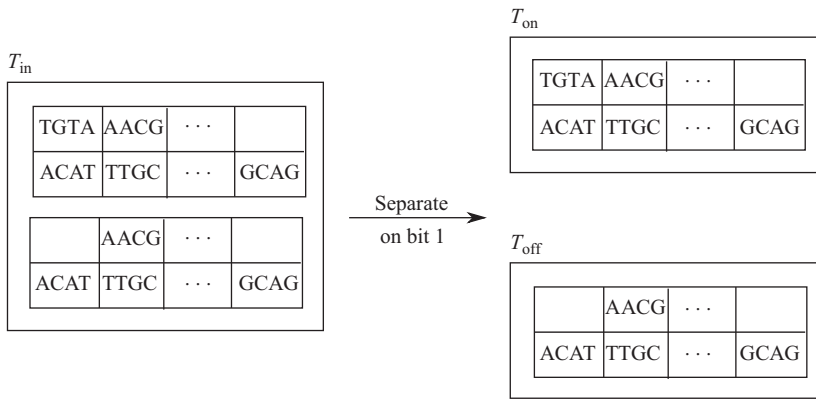


Figure 12.14 Separate operation

3. ***Set***: In this operation, a particular bit in every string of the DNA solution is set (turned on). The sticker for that bit is annealed to the appropriate region on every complex in the set's tube. Setting is performed by annealing the particular sticker to the bit needs to be set in the memory strand. Setting multiple bits can be done in parallel by pouring all the stickers corresponding to the targeted bits and annealed. All the poured stickers anneal with the memory strand at the same time.
4. ***Clear***: In this operation, a particular bit in every string of a DNA solution is removed by removing the sticker (if present) for that bit from every memory complexes in the test tube. Implementation of this operation is difficult, and hence avoided in proposed algorithms.

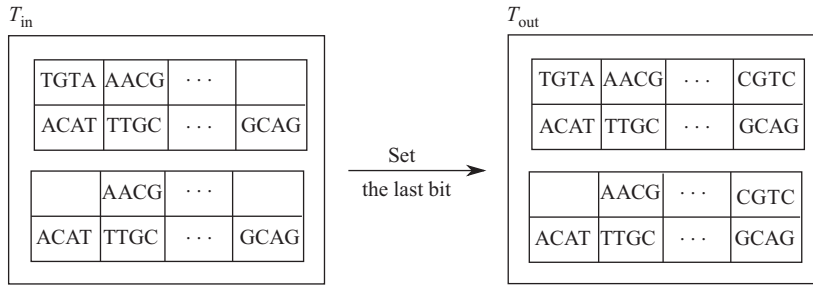


Figure 12.15 Set operation

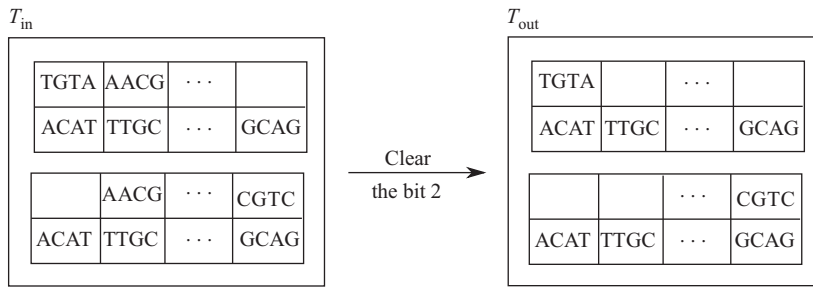


Figure 12.16 Clear operation

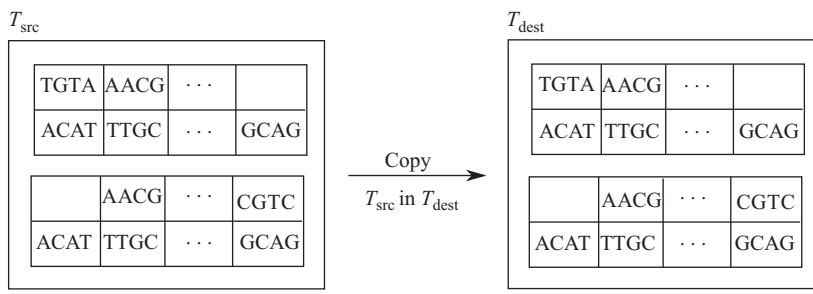


Figure 12.17 Copy operation (T_{dest} empty initially)

5. ***Copy***: In this operation, the contents of one test tube T_{src} are copied into another blank test tube T_{dest}. The contents of T_{src} are retained as it is. This operation can be performed easily by pouring some of the contents of T_{src} in T_{dest} and performing *Polymerase Chain Reaction* on both the test tubes. Difference between *Combine* and *Copy* is that, *Combine* empties the source test tube, but *Copy* retains, and *Copy* requires a blank test tube as destination.

12.2.2.2 Functional formulation of the operations available on Sticker-based DNA model

For the ease of representation, functional formulations of the operations described in Section 12.2.2.1 are used as follows:

1. *Copy* $\left(T^1_{dest}, T^2_{dest}, \ldots, T^n_{dest}; T_{src}\right)$: Copy contents of T_{src} in blank test tubes T_{dest}s.
2. *Combine* $\left(T_{dest}; T^1_{src}, T^2_{src}, \ldots, T^n_{src}\right)$: Pour the contents of T_{src}s in T_{dest}. After this operation, T_{dest} contains the union of contents of T_{dest} and all T_{src}s. T_{src}s become empty.
3. *Separate* $\left(T_1, i, b_{on}, b_{off}\right)$: Separate the contents of T_1 based on the value of ith bit in two test tubes b_{on}, containing DNA strands having ith bit on, and b_{off}, containing DNA strands having ith bit off.
4. *Set*(T_1, b_i): Set the ith bit b_i of all DNA strands in test tube T_1.

12.3 Performing arithmetic and logic operations using DNA

As has been observed till now, DNA computing is extremely useful and powerful for computationally "hard" problems that require very large search space. But it should be applicable on wider range of problems to make it generally applicable. To achieve that goal, simple logic and arithmetic operations available on a conventional computers are also necessary to be implemented in a DNA computer. These operations include different logic operations, viz. NOT, OR, AND, XOR, NOR, NAND, and XNOR; compare, shift, etc., integer and floating point arithmetic operations (Addition, Subtraction, Multiplication, and Division). Proper and efficient implementations of these operations can only lead DNA computing to be a complete substitute of conventional computers.

Several approaches have been proposed to perform arithmetic and logic operations using DNA in the literature starting from the first attempt to add two binary numbers by Guernieri et al. in 1996 [38] to several other attempts such as Gupta et al. [39] used fixed bit encoding scheme to perform arithmetic and logic operations, de Santis and Iaccarino [40] used different strands to represent each bit of a number. Several other approaches for implementing arithmetic and logic operations have been proposed, such as Ogihara–Ray [41] method of Boolean circuit simulation, Amos–Dunne [42] method, Barua–Misra [43] method, Qiu–Lu [44] method, etc.

One approach to implement arithmetic and logic operations using sticker-based DNA model is being described in this section. The operations being described here are parallel in time. Let two test tubes T_1 and T_2 be containing DNA strands corresponding to the two n-bit binary numbers under operation, respectively. The test tube T_{out} will contain the output after corresponding operation. At the beginning of each operation, T_{out} is considered to contain blank memory strands, i.e., the string $\underbrace{00\ldots00}_{n \text{ bits}}$. The implementations of some of the logic operations are described below.

12.3.1 AND operation

Algorithm 12.2 performs AND operation between two numbers represented using sticker model and are kept in two test tubes T_1 and T_2, respectively. The output of this operation is available in test tube T_{out}, also in sticker model representation.

Algorithm 12.2 Parallel AND logic

1: **procedure** AND($in : T_1, T_2; out : T_{out}$)
2: Copy$\left(T_1^{temp}; T_1\right)$.
3: Copy$\left(T_2^{temp}; T_2\right)$.
4: Combine$\left(T_C; T_1^{temp}, T_2^{temp}\right)$.
5: Copy$\left(T_C^1, T_C^2, \ldots, T_C^n; T_C\right)$.
6: **for all** bit b_i, **in parallel do**
7: Separate$\left(T_C^i, b_i, b_{on}, b_{off}\right)$.
8: **if** b_{off} is empty **then**
▷ b_i of both numbers is 1
9: Take sticker corresponding to b_i in test tube T_{Si}.
10: **end If**
11: **end for**
12: Pour all T_{Si}s in T_{out} and anneal.
13: **end procedure**

As an example, let us assume two numbers, 1001 and 1100 be kept in two test tubes T_1 and T_2, respectively. Let the memory strand be CGAT|GTAA|CCGG|CAGA. Hence, the two numbers will be represented as in Figure 12.18.

Now, the algorithm runs as follows:

1. The backup of T_1 and T_2 is combined into T_C. Hence, T_C now contains {1001, 1100}. Empty memory strand is taken into test tube T_{out}.

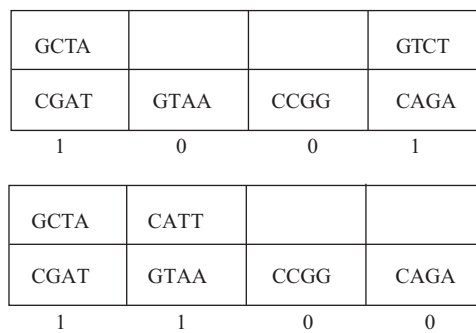


Figure 12.18 DNA molecules for the example of AND operation

GCTA			
CGAT	GTAA	CCGG	CAGA
1	0	0	0

Figure 12.19 DNA molecule representing output of AND operation

2. T_C is copied into four test tubes $T_C^1, T_C^2, T_C^3, T_C^4$.
3. The following operations are executed on the four test tubes in parallel:
   a. T_C^1 is separated based on bit 1, b_{on} contains both 1001 and 1100, and b_{off} is empty. So, the sticker for bit 1, *GCTA* is taken into test tube T_{S1}.
   b. T_C^2 is separated based on bit 2, b_{on} contains 1100, and b_{off} contains 1001. As b_{off} is not empty, nothing to do.
   c. T_C^3 is separated based on bit 3, b_{on} is empty, and b_{off} contains both 1001 and 1100. As b_{off} is not empty, nothing to do.
   d. T_C^4 is separated based on bit 4, b_{on} contains 1001, and b_{off} contains 1100. As b_{off} is not empty, nothing to do.
4. Pour T_{S1} in the T_{out} and anneal. T_{out} now contains the DNA strand as in Figure 12.19 representing the value 1000.

12.3.2 OR operation

Algorithm 12.3 performs OR operation between two numbers kept in two test tubes T_1 and T_2, respectively. The output of this operation is available in test tube T_{out}.

Algorithm 12.3 Parallel or logic

```
1: procedure OR(in : T_1, T_2; out : T_out)
2:     Copy(T_1^temp; T_1).
3:     Copy(T_2^temp; T_2).
4:     Combine(T_C; T_1^temp, T_2^temp).
5:     Copy(T_C^1, T_C^2, ..., T_C^n; T_C).
6:     for all bit b_i, in parallel do
7:         Separate(T_C^i, b_i, b_on, b_off).
8:         if b_on is not empty then
                                          ▷ b_i of at least one number is 1
9:             Take sticker corresponding to b_i in test tube T_Si.
10:        end if
11:    end for
12:    Pour all T_Sis in T_out and anneal.
13: end procedure
```

Algorithm 12.4 Parallel XOR logic

1: **procedure** XOR ($in : T_1, T_2; out : T_{out}$)
2: Copy$\left(T_1^{temp}; T_1\right)$.
3: Copy$\left(T_2^{temp}; T_2\right)$.
4: Combine$\left(T_C; T_1^{temp}, T_2^{temp}\right)$.
5: Copy$\left(T_C^1, T_C^2, \ldots, T_C^n; T_C\right)$.
6: **for all** bit b_i, **in parallel do**
7: Separate$\left(T_C^i, b_i, b_{on}, b_{off}\right)$.
8: **if** b_{on} is not empty **then**
▷ b_i of the two numbers are different
9: Take sticker corresponding to b_i in test tube T_{Si}.
10: **end if**
11: **end for**
12: Pour all T_{Si}s in T_{out} and anneal.
13: **end procedure**

12.3.3 XOR operation

Algorithm 12.4 performs XOR operation between two numbers kept in two test tubes T_1 and T_2, respectively. The output of the operation is available in test tube T_{out}.

12.3.4 NOT operation

Algorithm 12.5 performs NOT of a number kept in the test tube T_1. The output of this operation is available in test tube T_{out}.

Implementations of other logic operations, such as NOR, NAND, and XNOR can be implemented in similar fashion. Using these logic implementations, all integer operations, such as Comparator, Left and Right Shifters (Logical, Arithmetic, and Circular), Adder, Subtractor, Multiplier, and Divider; all floating-point arithmetic operations, such as Adder, Subtractor, Multiplier, and Divider can be implemented. Moreover, the floating point can be represented in IEEE 754 floating-point format.

As an example, let us implement the Comparator operation.

12.3.5 Comparator

The comparator operation 12.6 will compare two numbers in T_1 and T_2, and store the result in three test tubes T_g, T_l, and T_e, initially empty. If unequal, T_g will contain the greater number, T_l will contain the smaller number, and T_e will remain empty. If the two numbers are equal, T_e will contain both of the numbers, and the rest of these two test tubes remains empty.

Algorithm 12.5 Parallel NOT logic

1: **procedure** NOT ($in : T_1; out : T_{out}$)
2: Copy$\left(T_C^1, T_C^2, \ldots, T_C^n; T_1\right)$.
3: **for all** bit b_i, **in parallel do**
4: Separate$\left(T_C^i, b_i, b_{on}, b_{off}\right)$.
5: **if** b_{on} is not empty **then**
▷ b_i is 0
6: Take sticker corresponding to b_i in test tube T_{Si}.
7: **end if**
8: **end for**
9: Pour all T_{Si}s in T_{out} and anneal.
10: **end procedure**

Algorithm 12.6 Comparator

1: **procedure** COMPARATOR($in : T_1, T_2; out : T_g, T_l, T_e$)
2: Combine($T_1; T_2$).
3: **for all** bit b from MSB to LSB
4: Separate$\left(T_1, b, b_{on}, b_{off}\right)$.
5: **if** b_{on} or b_{off} is empty **then**
6: Combine$\left(T_1; b_{on}, b_{off}\right)$.
7: **else**
8: Combine $\left(T_g; b_{on}\right)$.
9: Combine$\left(T_l; b_{off}\right)$.
10: **break** ▷ Exit from for loop
11: **end if**
12: **end for**
13: **if** both T_g and T_l are empty **then**
▷ T_1 contains both strands
14: Combine($T_e; T_1$).
15: **end if**
16: **end procedure**

12.4 Implementing data structures using DNA

As has been described before, to make DNA computer generally applicable, those problems, which are very much implementable on conventional computer, should also be implementable on DNA computer. But to solve these problems, several abstract data types and techniques associated with them are unavoidable. These data structures define the data storage in proper fashion, for better applicability. Li et al. have implemented stack and queue in their work.

In this section, another approach to implement stack, queue, list, and map has been described.

12.4.1 Stack and queue using DNA

A stack may be considered as a last-in-first-out (LIFO) data structure with only a single entry and exit point named *top*. All elements are pushed into the stack through top, and also popped through the top. Hence, the elements which enter last, exit first.

On the other hand, a queue is a first-in-first-out (FIFO) data structure with two points, one for entry of elements, and the other one for exit. Element that enters first is also removed first.

For implementations, it may be assumed, without any loss of generality, that the DNA encoded strands of the elements are of equal lengths. To assure this, a restriction site is appended at the end of all DNA strands via ligation enzyme. Shorter DNA strands can be made of equal length with the longest strand by ligating additional dummy nucleotide bases after the restriction site.

As for example, there are m elements to be inserted into the data structure, and they are encoded as DNA strands $E_1, E_2, \ldots, E_m$ with E_s being the longest DNA strand. Let us assume, some fixed restriction site R_1R_2 is ligated at the end of all DNA strands, making the length of ith strand as $(E_i + R)$ for $1 \leq i \leq m$, where R is the length of restriction site R_1R_2. To assure that, all DNA strands have the same length, dummy DNA strand of length $(E_s - E_j)$ is ligated at the end of jth DNA strand for $1 \leq j \leq m$ and $j \neq s$.

12.4.1.1 Insertion into stack and queue

For the implementation of data structures, some unique DNA strand P is needed to be available in some test tube. The data structure is assumed to be built up in test tube T. Now, the insertion of element in the data structures can be performed using Algorithm 12.7.

As an example, if the elements $E_1, E_2, \ldots, E_m$ are inserted into the data structure one by one, according to the insertion algorithm, the elements will be inserted as follows:

- Prior to the addition of the first element E_1, T was empty. So simply pour E_1 into T. So, after this addition, $T = \{E_1\}$.
- When the next element E_2 will be added, T already contains E_1. Ligate P with the elements of T and add E_2. So, after this operation, $T = \{E_2,\ E_1P\}$.

Algorithm 12.7 Inserting element into data structure (*push* for Stack and *insert* for Queue)

1: **procedure** INSERT(*TestTube* T, *Element* E_i)
2: **if** T is not empty **then**
3: Add P into T and ligate.
4: **end If**
5: Pour E_i into T.
6: **end procedure**

- Similarly, after adding the third element, the elements of T will be $T = \{E_3, E_2P, E_1P^{(2)}\}$.

In this manner, after adding all of the m elements, elements of test tube T will be $T = \{E_m, E_{m-1}P, E_{m-2}P^{(2)}, \ldots, E_2P^{(m-2)}, E_1P^{(m-1)}\}$, where $P^{(i)}$ represents concatenation of i number of DNA strands P, i.e., $\underbrace{PPP\cdots P}_{i}$. So, the length of ith element inserted into the data structure becomes $[E + (i-1)P]$, where E is the length of the elements after adding the restriction site R_1R_2 and equalized by adding dummy nucleotide bases, and P is the length of the DNA strand P.

12.4.1.2 Removing element from data structure

Removing an element from the top of the stack is a trivial operation. As the length of the DNA strands decreases towards the top of the stack, the top element can be removed by simply performing Gel Electrophoresis and separating the band corresponding to the DNA strand having the smallest length.

Similarly, an element can be removed from the end of the queue by simply separating the band corresponding to the DNA strands having largest length.

After separating the desired band, the original element can be obtained using the following procedure:

1. The obtained DNA strand is annealed with the $\overline{R_1R_2}$, the Watson–Crick complement of the restriction site R_1R_2.
2. The annealed strand is cut at the proper location using the restriction enzyme, which detects the restriction site R_1R_2.
3. The solution is heated to denature.
4. The DNA strand containing R_1 is separated. The output contains the original element with R_1 attached at the end.

12.4.2 List using DNA

List can be implemented in the similar manner as stack and queue, but to insert element at any location other than the front, Algorithm 12.7 needs to be modified.

Assume, after inserting m elements, content of the test tube is as follows:

$$T = \left\{E_1P^{(m-1)}, E_2P^{(m-2)}, E_3P^{(m-3)}, \ldots, E_iP^{(m-i)}, E_{i+1}P^{(m-i-1)}, \ldots, E_{m-1}P, E_m\right\},$$

with 1 being the beginning index having the element E_1. Now, if some element E_j needs to be inserted into the list at index i, E_j needs to appended with $P^{(m-i)}$, and all the elements before the ith index need to have an additional P at the end. To obtain this, the following sequence of operations needs to be carried out:

1. The length of the DNA strand at index i is $[E + (m - i) * P]$ and all the elements at index $\leq i$ need to be appended with an additional P. So, perform Gel Electrophoresis and separate all the bands having length higher than or equal to the length $[E + (m - i) * P]$, into another test tube T_h. The rest of the solution is kept as it is in T.
2. Pour P into T_h and ligate using ligation enzyme to append P at the end of all the DNA strands.
3. Append P at the end of E_j $(m - i)$ number of times and keep in a test tube T_e.
4. Pour T_e and T_h back in T.

The final contents of the test tube T thus become as follows:

$$T = \{E_1P^{(m)}, E_2P^{(m-1)}, E_3P^{(m-2)}, \ldots, E_iP^{(m-i+1)}, E_jP^{(m-i)}, E_{i+1}P^{(m-i-1)}, \ldots, E_{m-1}P, E_m\},$$

having E_j inserted at the desired position.

Removal of an element from the list is same as in the removal of element from a stack or queue. Only difference is that, during Gel Electrophoresis, rather than separating the band having maximum (*remove* from queue) or minimum length (*pop* from stack), the band of the DNA strands with length $[E + (m - i) * P]$ needs to be separated to remove the element at index i.

12.4.3 Map using DNA

Map is an abstract data type and is a collection of (*key*, *value*) pair, where each key is unique and the retrieval of a value is performed using the key.

Let $(K_1, V_1), (K_2, V_2), \ldots, (K_m, V_m)$ be the pairs to be inserted into the map. Now, let, $E_{K1}, E_{K2}, \ldots, E_{Km}$ be the DNA strands encoding the keys $K_1, K_2, \ldots, K_m$, and $E_{V1}, E_{V2}, \ldots, E_{Vm}$ be the DNA strands encoding the values $V_1, V_2, \ldots, V_m$, respectively. The (*key*, *value*) pair (K_i, V_i) will be formed as the single DNA strand $E_{Ki}R_1R_2E_{Vi}$.

The possible operations on a map, viz., *reassign*, *remove*, and *lookup* operations are implemented as specified in the next three subsections.

12.4.3.1 Lookup

In *lookup* operation, the value of a pair (K_s, V_s) is retrieved using the value of the key K_s. This retrieval is performed by simply separating the DNA strand containing the string E_{Ks}. This separation by string can be performed using Watson–Crick complement of E_{Ks} on a magnetic bead, and letting the required DNA strand annealed with

the bead. The bead is then washed into another solution, and the strands are denatured by heating up. The solution thus contains the DNA strands $E_{Ks}\ R_1R_2E_{Vs}$. To retrieve the value V_s only, the resulting solution is annealed with $\overline{R_1R_2}$, cut at the specific location using the restriction enzyme, and separating the strands containing R_2.

12.4.3.2 Remove

In *remove* operation, some particular pair (K_s, V_s) is removed from the map. The implementation of this operation is trivial, viz., simply by performing the magnetic bead separation using the Watson–Crick complement of the given key E_{Ks}.

12.4.3.3 Reassign

To *reassign* the value of some particular pair (K_s, V_s) to the value of V_t, the required pair is first separated using the magnetic bead separation with the Watson–Crick complement of the key E_{Ks}, and kept in the test tube T_s. The separated pair is then annealed with $\overline{R_1R_2}$, and cut at the specific location using the restriction enzyme. The strand is cut into two parts, $E_{Ks}R_1$ and R_2E_{Vs}. The R_2E_{Vs} part is removed from the solution using magnetic bead containing $\overline{R_2}$.

In another test tube T_t, R_2 is kept, and the DNA strand E_{Vt} is appended at the end of R_2 using ligation enzyme. The solution in T_t is then poured in T_s and ligated using ligation enzyme again. The resulting solution in T_s will be formed as $E_{Ks}R_1R_2E_{Vt}$. This solution is then poured back in the main test tube containing the map.

12.5 Conclusion

In the present day scenario, conventional silicon computing is approaching a barrier whereas other computational techniques are emerging. Among these new techniques, computing using DNA molecules is proven to be powerful, especially for those problems, which need the complete search space to be searched, and hence "hard". Also, the extremely high data density of DNA molecules also proves it to be most useful for storing huge amount of data. So, all these advantages prove that in very near future if the bio-molecular operations can be automated, an inherently parallel powerful machine capable of storing huge data and performing search over huge search space is possible to be built.

References

[1] I. Miko. "Gregor mendel and the principles of inheritance". *Nature Education*, 1(1):134, 2008.

[2] R. Dahm. "Discovering DNA: Friedrich Miescher and the early years of nucleic acid research". *Human Genetics*, 122(6):565–581, 2008.

[3] J. Harwood. "The rediscovery of Mendelism in agricultural context: Erich von Tschermak as plant-breeder". *Comptes Rendus de l'Académie des Sciences-Series III-Sciences de la Vie*, 323(12):1061–1067, 2000.

[4] Charles Lenay. "Hugo de Vries: from the theory of intracellular pangenesis to the rediscovery of mendel". *Comptes Rendus de l'Académie des Sciences-Series III-Sciences de la Vie*, 323(12):1053–1060, 2000.

[5] E. von Tschermak-Seysenegg. "The rediscovery of Gregor Mendel's work: an historical retrospect". *Journal of Heredity*, 42(4):163–171, 1951.

[6] M. McCarty. "The Transforming Principle: Discovering that Genes Are Made of DNA". W.W. Norton & Company, New York, NY, 1986.

[7] O. T. Avery, C. M. MacLeod, and M. McCarty. "Studies on the chemical nature of the substance inducing transformation of pneumococcal types induction of transformation by a desoxyribonucleic acid fraction isolated from pneumococcus type iii". *The Journal of Experimental Medicine*, 79(2):137–158, 1944.

[8] C. O'Connor. "Isolating hereditary material: Frederick Griffith, Oswald Avery, Alfred Hershey, and Martha Chase". *Nature Education*, 1(1):105, 2008.

[9] A. Serafini. "Biology in the twentieth century". In *The Epic History of Biology*, pages 269–289. Springer, Berlin, 1993.

[10] J. D. Watson and F. H. C. Crick. "The structure of DNA". In *Cold Spring Harbor Symposia on Quantitative Biology*, volume 18, pages 123–131. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, 1953.

[11] J. D. Watson, F. H. C. Crick, "Molecular structure of nucleic acids". *Nature*, 171(4356):737–738, 1953.

[12] N. Goldman, P. Bertone, S. Chen, *et al.*, "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA". *Nature*, 494:77–80, February 2013.

[13] DNA data storage lasts thousands of years. http://news.discovery.com/tech/biotechnology/dna-data-storage-lasts-thousands-of-years-150817.htm. Accessed: 2015-11-02.

[14] G. M. Church, Y. Gao, and S. Kosuri. "Next-generation digital information storage in DNA". *Science*, 337(6102):1628, 2012.

[15] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark. "Robust chemical preservation of digital information on DNA in silica with error-correcting codes". *Angewandte Chemie International Edition*, 54(8):2552–2555, 2015.

[16] G. Luque and E. Alba. "Metaheuristics for the DNA fragment assembly problem". *International Journal of Computational Intelligence Research*, 1(2):98–108, 2005.

[17] Y. Zhang, B. Fu, and X. Zhang. "DNA cryptography based on DNA fragment assembly". In *2012 Eighth International Conference on Information Science and Digital Content Technology (ICIDT)*, volume 1, Jeju Island, Korea, pages 179–182, June 2012.

[18] A. Leier, C. Richter, W. Banzhaf, and H. Rauhe. "Cryptography with DNA binary strands". *Biosystems*, 57(1):13–22, 2000.

[19] G. Cui, C. Li, H. Li, and X. Li. "DNA computing and its application to information security field". In *Fifth International Conference on Natural Computation, 2009 (ICNC'09)*, volume 6, pages 148–152, August 2009.

[20] J. Chen. "A DNA-based, biomolecular cryptography design". In *Proceedings of the 2003 International Symposium on Circuits and Systems, Bangkok, Thailand, 2003 (ISCAS'03)*, volume 3, pages III-822–III-825, May 2003.

[21] K. Tanaka, A. Okamoto, and Isao Saito. "Public-key system using DNA as a one-way function for key distribution". *Biosystems*, 81(1):25–29, 2005.

[22] X. Wang and Q. Zhang. "DNA computing-based cryptography". In *Fourth International Conference on Bio-Inspired Computing, 2009 (BIC-TA'09)*, Beijing, China, pages 1–3, October 2009.

[23] M. Borda and O. Tornea. "DNA secret writing techniques". In *2010 Eighth International Conference on Communications (COMM)*, Bucharest, Romania, pages 451–456, June 2010.

[24] L. Liu, Q. Zhang, and X. Wei. "A RGB image encryption algorithm based on DNA encoding and chaos map". *Computers & Electrical Engineering*, 38(5):1240–1248, September 2012.

[25] R. K. Jangid, N. Mohmmad, A. Didel, and S. Taterh. "Hybrid approach of image encryption using DNA cryptography and TF Hill Cipher algorithm". In *2014 International Conference on Communications and Signal Processing (ICCSP)*, Tamilnadu, India, pages 934–938, April 2014.

[26] L. M. Adleman. "Molecular computation of solutions to combinatorial problems". *Science*, 266:1021–1024, November 1994.

[27] R. J. Lipton. "DNA solution of hard computational problems". *Science*, 268: 542–545, April 1995.

[28] D. Boneh, C. Dunworth, and R. J. Lipton. "Breaking DES using a molecular computer". In *DIMACS Workshop on DNA Computing*, Princeton, NJ, USA, 1995.

[29] Q. Ouyang, P. D. Kaplan, S. Liu, and A. Libchaber. "DNA solution of the maximal clique problem". *Science*, 278:446–449, October 1997.

[30] R. S. Braich, N. Chelyapov, C. Johnson, P. W. K. Rothemund, and L. M. Adleman. "Solution of a 20-variable 3-SAT problem on a DNA computer". *Science*, 296:499–503, April 2002.

[31] Y. Liu, J. Xu, L. Pan, and S. Wang. "DNA solution of a graph coloring problem". *Journal of Chemical Information and Computer Sciences*, 42:524–528, May–June 2002.

[32] C. A. A. Sanches and N. Y. Soma. "A polynomial-time DNA computing solution for the bin-packing problem". *Applied Mathematics and Computation*, 215:2055–2062, 2009.

[33] Computer made from DNA and enzymes. http://news.nationalgeographic.com/news/2003/02/0224_030224_DNAcomputer.html. Accessed: 2014-11-05.

[34] Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, and E. Shapiro. "An autonomous molecular computer for logical control of gene expression". *Nature*, 429:423–429, May 2004.

[35] Stanford creates biological transistors, the final step towards computers inside living cells, March 2013.

[36] G. I. Livshits, A. Stern, D. Rotem, *et al.*, "Long-range charge transport in single G-quadruplex DNA molecules". *Nature Nanotechnology*, advance online publication, 9(12): 1040–1046, October 2014.

[37] Israeli scientists achieve breakthrough in DNA computing, October 2014.

[38] F. Guernieri, M. Fliss, and C. Bancroft. "Making DNA add". *Science*, 273:220–223, 1996.

[39] V. Gupta, S. Parthasarathy, and M. J. Zaki. "Arithmetic and logic operations with DNA". In *Third DIMACS Workshop on DNA Computing*, Philadelphia, PA, 1997.

[40] F. de Santis and G. Iaccarino. "A DNA arithmetic logic unit". *WSEAS Transactions on Biology and Biomedicine*, 1:436–440, 2004.

[41] M. Ogihara and A. Ray. "Simulating Boolean circuits on a DNA computer". In *Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB'97)*, Santa Fe, NM, USA, pages 226–231, 1997.

[42] M. Amos and P. E. Dunne. "DNA simulation of Boolean circuits". Technical report. In *Proceedings of Third Annual Genetic Programming Conference*, Stanford, California, USA, 1997.

[43] R. Barua and J. Misra. "Binary arithmetic for DNA computers". In *Revised Papers from the Eighth International Workshop on DNA Based Computers: DNA Computing*, Sapporo, Japan, DNA8, pages 124–132, 2002.

[44] Z. F. Qiu and M. Lu. "Arithmetic and logic operations for DNA computers". In *Proceedings of the Second IASTED International Conference on Parallel and Distributed Computing and Networks*, Brisbane, Australia, pages 481–486, 1998.

Index

acoustic phonons 182–3
adaptive routing 348, 365
adenine 380
adenosine triphosphate (ATP) 383
Adleman–Lipton model 388
  solving Hamiltonian path problem 388–92
  solving SAT problem 392–3
aging variations 150
analytical modeling of RTL components 291–3
AND operation 392, 398–9
ANSYS Mechanical software 307, 332
application-specific circuits 267
Application Specific Integrated Circuit (ASIC) 259, 268
As Late As Possible (ALAP) schedule 227–8, 281, 283–4
As Soon As Possible (ASAP) schedule 227–8, 281–4
asymmetrically gate overlap/underlap engineered FinFET (FinFET-Asym1) 117–21, 129–30, 135
asymmetrically gate-underlap engineered FinFET (FinFET-Asym2) 121–4, 129, 132, 135
asymmetrical SRAM cell (SRAM-Asym) 119–21, 124–5, 128–37
automated design methods 220
AutoPilot 259
auto-regressive filter (ARF) 293, 297
auxiliary bacterium 243
  encoding of 245–6
average dynamic power 231, 238–9, 257

back-gate bias IG-FinFET-based 6T SRAM 82–3
back-gate bias PPN 10T SRAM 85–6
back-gate bias voltage (BGV) 97, 99, 102–5
Bacterial Foraging Optimization Algorithm (BFOA) 221, 227, 239, 241, 253–4, 261
  exploration process 243–4, 248–52
ball grid array (BGA) package 47
band-pass filter (BPF) 293
Bayesian model fusion (BMF) 3, 11–12, 24–6
  post-silicon indirect sensor calibration via: *see* post-silicon indirect sensor calibration: via Bayesian model fusion
Bayes' theorem 15
bias current 21, 29–30, 187, 192–3
bias voltage, optimal 21, 23
binary instrumentation based method 364
bio/nature-inspired algorithms for DSE framework 241
  Bacterial Foraging Optimization Algorithm (BFOA) 241
  BFOA-exploration process 243–4, 248–52
  encoding/initialization of datapath bacterium 244–5

encoding of auxiliary bacterium 245–6
models for metric 248
particle swarm optimization (PSO) 241–3
proposed movement of bacterium 246–8
body biasing 63–5, 150
Boltzmann transport equation (BTE) 185, 196
brute-force search algorithm 19–21
bulk FinFET 73–5
buried oxide (BOX) 72

CAMAD design systems 227
carbon nanotubes (CNTs) 201, 346
-based interconnects 173, 175–6
electrical modeling of 202–5
electrostatic coupling 204
quantum capacitance 204
shell-to-shell coupling capacitance 205
thermal modeling for 205–16
carbon nanotube VLSI interconnects 173
electrical properties 176
effective mean free path 182–4
equivalent capacitance 181–2
equivalent circuit 184
equivalent inductance 180–1
equivalent resistance 177–9
present status of VLSI interconnect 174–5
survey of CNT-based interconnects 175–6
thermal properties 185
iterative scheme for resistance and temperature 190–1
MWCNT 189–90
performances in terms of *S*-parameters 193–6
SWCNT bundle 187–8
SWCNTs 185–7
temperature profiling inside interconnect 191–3
CatapultC 259
C/C++/systemC 259
C/C++/systemC-to-RTL synthesis 259
cell stability, defined 88
cell-to-cell delay mismatch 41
charge pump circuit 155–6
chemotaxis mechanism 247, 254
classic nano-CMOS based circuits and systems 273
clock distribution network (CDN) 323–6
CMP (chip multi-processor system) 338
Colpitts voltage-controlled oscillator (VCO) 3
comparator 99, 400
complementary metal-oxide-semiconductor (CMOS)
-based SRAMs 71
VLSI fabrication technologies 142
complex digital integrated circuits 267
computer-aided design (CAD) methods 220
concurrent error detection (CED) technique 252
constant field scaling 144–5
constant voltage scaling 144
control data flow graph (CDFG) 220, 228, 239, 244
conventional six-FinFET SRAM cell 114–16
cost model 257
cross-coupled back-gate bias PPN 10T SRAM 86–7
Crosstalk Avoidance Codes (CAC) 344, 365
cycle accurate simulation 363
cycle specific simulation 363
cytosine 380

data access speed 131–2
data flowgraph (DFG) 272
datapath bacterium 243
encoding/initialization of 244–5

data stability and write ability enhancement techniques for FinFET SRAM circuits 113
8Kbit memory arrays designed with different SRAM cells (case study) 128
data access speed 131–2
hold static noise margin (HSNM) 129–30
leakage power consumption 132–3
read static noise margins (RSNMs) 128–9
write voltage margin 130–1
fabrication and SRAM cell area comparison 127–8
six-FinFET SRAM cells 114
conventional six-FinFET SRAM cell 114–16
hybrid SRAM cell with asymmetrically overlapped/underlapped bitline access transistors 120–1
independent-gate FinFET SRAM cell 116–17
single-ended read SRAM cell with underlap engineered symmetrical-FinFETs 125–7
SRAM cell with asymmetrically gate-underlapped transistors 121–5
SRAM cell with asymmetrically overlap/underlap engineered FinFETs 117–20
variations of underlap lengths due to process imperfections 133–7
data structures implementation using DNA 402
list using DNA 403
map using DNA 404
lookup operation 404
reassign operation 405
remove operation 405
stack and queue using DNA 402
insertion into stack and queue 402–3
removing element from data structure 403
DC–DC converter design 316–17
experimental results 319–23
overview of 317
TSV inductor design 317–19
delay (execution time) model 257
delay locked loop (DLL)-based test structure 40
delay measurement
gate delay measurement cell 42–3, 57–8
inverting average gate delay measurement 45–8
measured accuracy of 53–5
non-inverting average gate delay measurement 44–5
delay variation 40, 49
due to layout orientation 52
due to supply voltage 52–3
Dennard's scaling theory 142–4
design space exploration (DSE) 224, 240, 269
design space exploration (DSE) framework, bio/nature-inspired algorithms for 241
Bacterial Foraging Optimization Algorithm (BFOA) 241
exploration process 243–4, 248–52
encoding/initialization of datapath bacterium 244–5
encoding of auxiliary bacterium 245–6
models for metric 248
particle swarm optimization (PSO) 241–3
proposed movement of bacterium 246–8
device scaling 142
constant field scaling 144–5
constant voltage scaling 144
generalized scaling 145
DGFET 73
DG-FinFET 73

die-to-die variations 35
diffusion effect, length of 51–2
digital integrated circuits 219
  abstraction levels 221
digital signal processing (DSP) 260, 267
digital-to-analog converter (DAC) 20
discrete-cosine transformation (DCT) filter 293
discrete particle swarm optimization (PSO) 227
DNA computing 379
  DNA solutions, operations on 381–5
    amplify 383–4
    anneal 383
    denaturation 383
    ligation 383
    separation by length 385
    separation by string 384
    synthesis 381
  DNA structure 380–1
  history of 386
    attempts made for physical realization 387
    emergence for hard problem solving 386–7
  implementing data structures using DNA 402
    list using DNA 403–4
    map using DNA 404
    stack and queue using DNA 402–3
  models 387–8
    Adleman–Lipton model 388
    sticker model 393
  performing arithmetic and logic operations 397
    AND operation 398–9
    comparator 400
    NOT operation 400
    OR operation 399
    XOR operation 400
  power of DNA computer 386
double-gate MOSFET (DGMOS) 72–3
double modular redundant (DMR) system 252, 254
drain induced barrier lowering (DIBL) 72
dual sleep approach 159
dual threshold CMOS (DTCMOS) technique 154
duty cycle, measurement of 61–2
dynamic power consumption 275, 277
dynamic power dissipation 145, 154–5, 267
dynamic power management (DPM) techniques 275
dynamic programming-based runtime thermal management (DPRTM) policy 348
Dynamic Voltage and Frequency Scaling (DVFS) 364
dynamic voltage scaling (DVS) 95, 102, 275

early-stage sensor model 12
effective mean free path 178, 182–5
electro-migration 150, 174–5
electronic design automation (EDA) tool 224, 267
electrostatic capacitance of CNT 181–2
electro-thermal coupled equations, iterative scheme for 191
elimination dispersal (ED) algorithm 246–7
elimination-dispersal mechanism 247–8
elliptic-wave filter (EWF) 293
energy dissipation 273
environmental and aging related variations 150
environmental variation 36
Error Correction Code (ECC) scheme 344, 365
event-driven simulation 362–3

fault models 364
field programmable gate arrays (FPGA) 259–60
filtering model: *see* Adleman–Lipton model

FinFET-based SRAM design challenges 91–2
FinFET-based SRAM topologies 79
  back-gate bias IG-FinFET-based 6T SRAM 82–3
  IG-FinFET-based 6T SRAM 80
    read and write operation 81
    SRAM cell design 81–2
  IG-FinFET-based PPN 10T SRAM 83
    back-gate bias PPN 10T SRAM 85–6
    cross-coupled back-gate bias PPN 10T SRAM 86
    hybrid back-gate bias PPN 10T SRAM 86–7
  stability analysis 88–91
finite impulse response (FIR) filter 293
flits 341, 349
floorplan driven multi-voltage synthesis 231–3
flow control mechanism 349
Force-Directed Scheduling (FDS) 227–8
forced NMOS technique 153
forced PMOS technique 153
forced stack technique 152–4
free space optical link based photonic links 355
full system simulation 363–4

GALEOR (GAted LEakage transistOR) technique 161–2
GALS (globally asynchronous locally synchronous) architecture 338
gate-controlled electric field 74
gate delay measurement cell (GDMC) 42–3, 57–8
gate delay measurement using reconfigurable ring oscillator (RRO)
  gate delay measurement cell 42–3
  RRO structure 43
    comparison with other works 55–6
    delay variation due to layout orientation 52
    delay variation due to supply voltage 52–3
    inverting average gate delay measurement 45–8
    length of diffusion (LOD) effect 51–2
    measured accuracy of the delay measurement 53–5
    measured results 48–9
    non-inverting average gate delay measurement 44–5
    poly-pitch effect 49–51
gate delay variability 39–41
gate induced drain leakage (GIDL) 74, 146, 273
gate-oxide leakage modeling 276, 287–9
gate-to-substrate coupling capacitance 74
gate under test (GUT) 40, 65–6
Gaussian distribution 13, 15, 61, 89, 99, 133
GAUT 260
Gel Electrophoresis 385, 391, 403–4
generalized scaling 145
General Purpose Graphics Processing Units (GPGPUs) 366
generic NoC 339
genesis behavioral synthesis system 229
genetic algorithm (GA)-based scheduling 227
global variation: *see* die-to-die variations
graphene 176, 204, 208
graphene-enabled wireless communications (GWC) system 346
graphene-enabled wireless NoC (GWNoC) 346
green on-chip inductors 305
  low-frequency applications 316
    DC–DC converter design 316–23

resonant clocking implementation 323–6
micro-channel shielding 326–34
parameters 307–9
design parameters 313
impact of process parameters 309
liner thickness 312
loop pitch 315–16
metal height 312–13
metal width 316
number of tiers 314–15
number of turns 313–14
substrate conductivity 310–11
substrate height 309–10
TSV diameter 311
guanine 380

HAL differential equation solver 293
handheld devices, design of 219–20
hardware memory management unit (HwMMU) 352
heat diffusion equation 196
high-level synthesis (HLS) 267–8
scheduling, allocation, and binding during 280
SPICEless RTL optimization during 271–2
high-level synthesis (HLS) of digital integrated circuits 219
allocation and binding 222, 228–9
bio/nature-inspired algorithms for DSE framework 241
Bacterial Foraging Optimization Algorithm (BFOA) 241
BFOA-exploration process 243–4, 248–52
encoding/initialization of datapath bacterium 244–5
encoding of auxiliary bacterium 245–6
models for metric 248
particle swarm optimization (PSO) 241–3
proposed movement of bacterium 246–8
design process 222–4
future directions of 260–1
need for 224–5
power, energy, or leakage aware HLS for nanoscale ICs 229
design space exploration approaches during HLS 239–40
effects of loop manipulation on power and delay of design 233–9
selected power, energy, or leakage aware HLS methods 229–32
scheduling algorithms 222, 225–8
constructive/iterative 227–8
transformational 226–7
for secure information processing 252
related work 252–3
results of exploration process of hardware Trojan secured datapath 258
security against untrusted third party digital IPs 253–7
tools available for 258
commercial tools 259–60
free HLS tools 260
high level transformations 220, 223, 260
hold static noise margin (HSNM) 129–30
hot-carrier injection (HCI) 150
leakage 146
hybrid back-gate bias PPN 10T SRAM 86–7
hybrid photonic NoC 356

IBM 142, 307, 329
independent-gate FinFET (IG-FinFET) 75, 78–9
independent-gate FinFET (IG-FinFET)-based 6T SRAM 80, 82–3
read and write operation 81
SRAM cell design 81–2

independent-gate FinFET (IG-FinFET)-based PPN 10T SRAM 83
  back-gate bias PPN 10T SRAM 85–6
  cross-coupled back-gate bias PPN 10T SRAM 86
  hybrid back-gate bias PPN 10T SRAM 86–7
independent-gate FinFET SRAM cell 116–17
independent variations 37
indirect performance sensing 2–4, 18
inductive DC–DC converters 307, 316
input pattern control technique 153
instruction set architecture (ISA) 363
integrated circuit (IC) 141, 149, 173, 219
interconnect 173
inter-die variation: *see* die-to-die variations
interleaved buck converter 317–18
International Technology Roadmap for Semiconductors (ITRS) 39, 114
intra-die variation: *see* within-die variation
inverter-to-inverter delay 48
inverting average gate delay measurement 45–8
inverting gate
  rise and fall delays of 60–1

Joule heating 174–6, 185–7, 189, 191–2, 195–6, 210

L_0-norm regularization 5–9
L_1-norm regularization 2, 5, 7–11
large-scale process variation 1
layout orientation, delay variation due to 52
LC resonant clocking 307, 323–5
leakage control transistors (LCTs) 158
leakage current components 146–7
leakage feedback technique 156–7
leakage-optimal digital integrated circuits 268–9
leakage power 146
leakage power consumption 113, 132–6
leakage power dissipation 141–2, 145–6, 154, 157, 273
leakage reduction techniques 150
  dual threshold CMOS (DTCMOS) technique 154
  forced stack technique 152–4
  GALEOR technique 161–2
  leakage feedback technique 156–7
  LECTOR technique 158
  MTCMOS technique 150–2
  SCCMOS (super cut-off CMOS) technique 154–6
  sleepy keeper technique 159–60
  sleepy stack technique 158–9
  variable threshold CMOS (VTCMOS) technique 157–8
  VCLEARIT technique 160–1
LECTOR (LEakage ControlTransistOR) technique 158, 164
Leeson's model 21
LegUp 260
length of diffusion (LOD) effect 51–2
line edge roughness (LER) 35, 38–9
liner thickness 312
list scheduling 227–8
local variation: *see* within-die variation
lookup operation 404
loop manipulation techniques 220, 233, 261
loop pipelining 220, 235–9, 251–2
loop pitch 315–17
loop shifting 220, 238
loop unrolling 220, 231, 233–5, 237–9, 249, 261
low-*k* dielectric 174–5
low-leakage techniques for nanoscale CMOS circuits 141
  device scaling 142
    constant field scaling 144–5

constant voltage scaling 144
generalized scaling 145
issue of leakage current 148
leakage analysis 162–6
leakage reduction techniques 150
dual threshold CMOS (DTCMOS) technique 154
forced stack technique 152–4
GALEOR technique 161–2
leakage feedback technique 156–7
LECTOR technique 158
MTCMOS technique 150–2
SCCMOS (super cut-off CMOS) technique 154–6
sleepy keeper technique 159–60
sleepy stack technique 158–9
variable threshold CMOS (VTCMOS) technique 157–8
VCLEARIT technique 160–1
power dissipation 145
leakage current components 146–7
leakage power dissipation 145–6
variability issues and aware design 148–50
low-noise amplifier (LNA) 3, 27–9
60GHz LNA (case study) 26–32
low power dissipation design 142
low power/energy HLS approaches 230
Luttinger liquid theory 204–5

magnetic inductance 180, 205
map using DNA 404
lookup operation 404
reassign operation 405
remove operation 405
Markov modulated Poisson processes (MMPP) 362
MATLAB 272
MATLAB®/Simulink® simulations 285
Matthiessen's Rule 182–3
maximum-a-posteriori (MAP) estimation 3
measurement accuracy 65
Mentor Graphics 259
metal height (h) 312–13
metallic CNTs (m-CNTs) 201
metal-oxide-semiconductor (MOS) 71, 141
Metal Oxide Semiconductor Field Effect Transistor (MOSFET) scaling 142–3, 174
metal width 316
micro-channel shielding 326–34
microelectro-mechanical systems (MEMS) 299
micro-ring resonator based photonic links 354
minimum leakage vector 153
mitochondrial DNA (mtDNA) 380
Mode of Operations (MOPs) 93
Monte-Carlo (MC) simulation results 72
Moore's law 142–3
MPSoC (multi-processor system on chip) 338, 353
multiple gate transistors 276
multi-threshold CMOS (MTCMOS) technique 150–2
multi-voltage CMOS (MVCMOS) technique 154–5
multi-wall carbon nanotubes (MWCNTs) 175–6, 201
circuit model of 203
equivalent circuit for 184
equivalent inductance of 180–1
equivalent resistance of 177–9
multiple shells of 203
resistance 203
resistance distribution for 211–12
temperature coefficient of resistance (TCR) for 208–11
thermal properties of 189–90
voltage drop for 215

nano-electro-mechanical systems (NEMS) 299
nanoelectronic technology 267–8
Nanonexus 307
nanoscale FinFET devices 72

bulk FinFET 73–5
SOI FinFET 75
independent-gate FinFET (IG-FinFET) 78–9
omega-gate FinFET (Ω-gate FinFET) 77–8
shorted-gate FinFET (SG-FinFET) 76–7
N-channel metal oxide semiconductor field effect transistor (NMOS) 51, 152
negative bias temperature instability (NBTI) 40, 62, 150
negative bitline (NBL)-driven design 95–7
network interface controller (NIC) 341
neutral length 206
NoC (network-on-chip) 337
basics 338
transition towards 3D 338–40
NoC (network-on-chip), three dimensional 337
architectural optimization of 349
interconnection 351
memory 352
network interface controller 350–1
router architecture 349–50
design challenges in 340–5
emerging technological challenges 345
macro-architecture 341
energy-aware modelling and design 343–4
fault tolerance 344
mapping application 343
micro-architecture 341
performance 342
physical design 342–3
reliability analysis 344
routing 341
thermal issues 344
topology 341
performance centric design of 347
adaptive routing 348
flow control mechanism 349
interconnection topology development 347
oblivious routing 348
routing policy 347–8
photonic 353
multi-dimensional design issues in 357–8
photonic interconnect for manycore ICs 353
reliability and fault tolerance in 3D NoCs 364–6
simulators 361
NoC simulation 361–4
simulators 361–4
thermal-aware design 352–3
wireless 358
inductive coupling interconnected application-specific 3D NoC 359–60
low-latency-based wireless 3D NoCs 358–9
reconfigurable hybrid 3D wireless NoC 360–1
NoC (network-on-chip), two dimensional 338
tile-based 2D mesh topology 339
noise figure (NF) 26–7, 30
non-inverting average gate delay measurement 44–5
non-inverting gate
rise and fall delays of 58–60
non-systematic variation 37
NOT operation 400
nucleotides 380
NUMA (non-uniform memory access) paradigm 352
number of tiers 314–15
number of turns 313–14

oblivious routing 348
omega-gate FinFET 77–8
on-chip gate delay variability measurement in scaled technology node 35
classification of variability 36–8

gate delay measurement cell (GDMC) 42–3, 57–8
gate delay variability 39–41
measured results 62
comparison with the existing techniques 65–6
impact of body-bias 63–5
impact of supply voltage 65
measurement accuracy 65
measurement of rise and fall delays using standard RO 56–7
reconfigurable ring oscillator (RRO) structure 43
comparison with other works 55–6
delay variation due to layout orientation 52
delay variation due to supply voltage 52–3
inverting average gate delay measurement 45–8
length of diffusion (LOD) effect 51–2
measured accuracy of the delay measurement 53–5
measured results 48–9
non-inverting average gate delay measurement 44–5
poly-pitch effect 49–51
rise and fall delays
of inverting gate 60–1
of non-inverting gate 58–60
rise and fall gate delay variability 41–2
sources of variability 38–9
test chip and measurement results 61
measurement of duty cycle 61–2
on-chip sampling oscilloscope 41, 66
on-chip self-healing 1–2, 4
on-chip self-healing flow 3, 17–20
Brute-force search for 20
one-dimensional Fourier heat equation 176
optical detector 353
optical filter 353
optical link 353
optical modulator 353
optical phonons 182–3, 209
optical source 353
optimal bias voltage 23
optimized hardware Trojan secured datapath, generation of
DSE methodology for 253
ordinary least squares (OLS) fitting method 5, 24
OR operation 399
oscillation amplitude 21
oscillation frequency 21
over-simplified model 4
oxide thickness variation (OTV) 35, 39

particle swarm optimization (PSO) 93, 231, 241–3
patterned ground shield (PGS) 307, 318
P-channel metal oxide semiconductor field effect transistor (PMOS) 51, 66, 152, 156, 159–60, 162
performance of interest (PoI) 2, 18
performances of measurement (PoMs) 2–3, 18
phase locked loop (PLL) 61–2
phonons 182, 196
photonic 3D NoC 346, 353
multi-dimensional design issues in 357–8
photonic interconnect for manycore ICs 353
photonic devices and systems 353
systems based on PNoC 355–6
photonic links 353
free space optical link based 355
micro-ring resonator based 354
quantum dot (QD) LED based 354–5
picosecond imaging circuit analysis (PICA) method 40
polymerase chain reaction (PCR) 383–4
poly-pitch effect 49–51

positive bias temperature instability (PBTI) 62, 150
post-silicon indirect sensor calibration 2–3
  via Bayesian model fusion 11
    MAP estimation 14–17
    prior knowledge definition 12–14
post-silicon measurement 4
power aware HLS 231
power dissipation 145, 273
  leakage current components 146–7
  leakage power dissipation 145–6
PPN 10T SRAM 83
  back-gate bias 85–6
  cross-coupled back-gate bias 86
  hybrid back-gate bias 86–8
Predictive Technology Model (PTM) 88
pre-silicon indirect sensor modeling 2–4
  via sparse regression 4
    L_0-norm regularization 5–7
    L_1-norm regularization 7–11
pre-silicon techniques 275
process, voltage and temperature (PVT)-aware SRAM design 92
  leakage-driven design 97
    identification of PCs 99–101
  negative bitline (NBL)-driven design 95–7
  PVT mitigation techniques 93
    dynamic mitigation techniques 94–5
    static mitigation techniques 93–4
  sensitivity-driven design 101–3
  stability analysis 103–5
process, voltage and temperature (PVT) variations 55, 164
process variation 1, 36, 149
propagation delay modeling 289–91

Q-fold cross-validation 6, 9
quantum capacitance 181–2
quantum dot (QD) LED based photonic links 354–5
quantum inductance 180

radio frequency (RF) circuits 306
random dopant fluctuations (RDFs) 35, 38, 76
random/independent variations 37
random sampling 41
rapid scaling of silicon technology 148
read noise margin (RNM) 80, 92, 104
read static noise margins (RSNMs) 128–9
reassign operation 405
reconfigurable ring oscillator (RRO)
  gate delay measurement using: *see* gate delay measurement using reconfigurable ring oscillator (RRO)
register-transfer level (RTL) structure 268
regular grid-based topology 347
regular PSO 227
remove operation 405
resonant clocking implementation 323
  experimental results 325–6
  inductor placement 325
  overview 323–5
resonant clocks 323–5
restricted isometry property (RIP) 10
ring oscillator (RO) 38, 40
rise and fall gate delay measurement using RRO 56–7
  gate delay measurement cell 57–8
  rise and fall delays of inverting gate 60–1
  rise and fall delays of non-inverting gate 58–60
rise and fall gate delay variability 41–2
ROCCC 260

SCCMOS (super cut-off CMOS) technique 154–6
scheduled DFG (SDFG) 280
scheduling algorithms used in HLS 222, 225–8
  constructive/iterative 227–8
  transformational 226–7

secure information processing, HLS approaches for 252
  related work 252–3
  results of exploration process of hardware Trojan secured datapath 258
  security against untrusted third party digital IPs 253–7
    evaluation models 256–7
    incorporating vendor allocation procedure 'v' in problem encoding 255–6
self-healing analog/RF circuits 1
  case study 20
    25GHz differential Colpitts VCO 20–6
    60GHz LNA 26–32
  indirect performance sensing 3–4
  on-chip self-healing flow 17–20
  post-silicon indirect sensor calibration via Bayesian model fusion 11
    MAP estimation 14–17
    prior knowledge definition 12–14
  pre-silicon indirect sensor modeling via sparse regression 4
    L_0-norm regularization 5–7
    L_1-norm regularization 7–11
self-resonant frequency (SRF) 307
semiconducting CNTs (s-CNTs) 201
shallow trench isolation (STI) 51
shorted-gate FinFET (SG-FinFET) 75–7
silicon-on-insulator (SOI) FinFET 75
  independent-gate FinFET 78–9
  omega-gate FinFET 77–8
  shorted-gate FinFET 76–7
silicon-on-insulator (SOI) technology 72
silicon photonics 353–4
Simscape 272
simulated annealing (SA) 227
Simulation Program with Integrated Circuit Emphasis (SPICE) model 36, 54–5
Simulink 272
single-ended read SRAM cell (SRAM-SR) with underlap engineered symmetrical-FinFETs 125–8, 132, 136–7
Single Event Upsets (SEUs) 365
single-phase buck converter 317–18
single wall carbon nanotube (SWCNT) 175–6, 201
  bundle of 176
    equivalent circuit for 184
    equivalent inductance of 180–1
    equivalent resistance of 177–9
    thermal properties of 187–8
  equivalent circuit for 184
  equivalent inductance of 180–1
  equivalent resistance of 177–9
  Fourier heat equation 185
  metallic 203
  resistance distribution for 208–9
  semiconducting 203
  thermal coefficient for resistance for 206–7
  thermal properties of 185–7
Single Write Multiple Read (SWMR) 356
six-transistor static random-access memory (6T SRAM) cell 80, 113
skip-links 358
sleepy keeper technique 159–60
sleepy stack technique 158–9
SMT (simultaneous multi-threading) 338
SoC (system-on-chip) 337, 352
Soft Error Rate (SER) 365
software pipelining: *see* loop pipelining
S-parameter calculation 195
SPARK tool 260
sparse regression (SR) 2, 29
  pre-silicon indirect sensor modeling via: *see* pre-silicon indirect sensor modeling: via sparse regression

spatially correlated variations 37
spatial process variations 149
SPICEless RTL design optimization 267
  during HLS 271–2
  experimental results for RTL optimization 293–8
  heuristic algorithm for 281–5
  objective function for 279–81
  overall RTL optimization flow 278–9
  of power dissipation in digital circuits 272–4
  power optimization at RTL 274
    existing methods for 274–6
    gate-oxide leakage optimization, multiple oxide thickness technology for 276–8
  of RTL component library 285–7
    analytical modeling of RTL components 291–3
    gate-oxide leakage modeling 287–91
SPICE simulations 269
stability analysis 88–91
static noise margin (SNM) 80
static power dissipation 145
static random-access memory (SRAM) cells 71, 113
  with asymmetrically gate-underlapped transistors 121–5
  with asymmetrically overlap/underlap engineered FinFETs 117–20
  with independent-gate bitline access transistors (SRAM-Inde) 116, 125, 135, 137
  SRAM-Hybrid1 120–1, 124–5, 127–37
  SRAM-Hybrid2 123–5, 127–37
  with tied-gate FinFETs (SRAM-Tied) 115, 121, 125, 128
sticker-based DNA 397
  clear operation 395–6
  combine operation 394–5
  copy operation 396
  separate operation 394–5
  set operation 395–6
sticker model 393
  functional formulation of operations available on 397
  sticker-based DNA, operations on 394–6
substrate conductivity 310–11
substrate height 309–10
sub-threshold leakage 146
supply voltage
  delay variation due to 52–3
  impact of 65
Swarm Intelligence (SI) 221
symmetrically gate-underlapped FinFETs (FinFET-Sym) 114–15, 118, 120, 122
systematic variation 37
SystemCoDesigner 260
system-on-chip (SoC) 20, 72
SystemVerilog 268, 272, 279

temperature coefficient of resistance (TCR) 206
temporal variations 149
test chip and measurement results 61
  measurement of duty cycle 61–2
thermal modelling 364
  for CNTs 205–16
thiamine 380
third party digital Intellectual Property (IP) cores (3PIPs) 221
three-dimensional integrated circuits (3D ICs) 305, 337, 358, 364
three-dimensional transistors: *see* multiple gate transistors
through-silicon-via (TSV) inductors 305, 342
tile-based 2D mesh topology 339
time dependent dielectric breakdown (TDDB) 150
toroidal TSV inductor 306–7, 319, 322
total power dissipation 145, 273
transistor device dimensions 142
tree-based topology 347

Trojans (hardware) 221, 252, 255, 261
TSV diameter 311
25 GHz differential Colpitts VCO (case study) 20–6

ultra large scale integration (ULSI) technology 141
unrolling factor (UF) 234–5
unscheduled DFG (UDFG) 280
untrusted third party digital IPs, security against 253–7

variability
  classification of 36–8
  sources of
    line edge roughness (LER) 38–9
    oxide thickness variation (OTV) 39
    random dopant fluctuations 38
variability issues and aware design 148–50
variability measurement
  gate delay variability 39–41
  rise and fall gate delay variability 41–2
variable threshold CMOS (VTCMOS) technique 157–8
variation, taxonomy of 36
VCLEARIT (VLSI CMOS LEAkage ReductIon Technique) technique 160–1
vendor allocation procedure 253
vendor allocation procedure type 'V' 254
  incorporating, in problem encoding 255–6
vertical cavity surface emitting LASER (VCSEL) diodes 355
vertical TSV inductor 320
very large scale integration (VLSI) circuits 36
very large scale integration (VLSI) interconnect
  heat source and sinks in 190
  present status of 174–5
Virtual-Channel (VC) 365
virtual machine (VM) 363
Virtual-Output-Queue (VOQ) 365
VLSI/CAD community 229, 239
voltage-controlled oscillator (VCO) 3, 21–2
voltage scaling techniques 275
voltage transfer characteristics (VTC) 125–6, 128

wavelength division multiplexed (WDM) 354
weighted sum particle sum optimization (WSPSO) 227
wireless 3D NoC 358
  inductive coupling interconnected application-specific 3D NoC 359–60
  low-latency-based wireless 3D NoCs 358–9
  reconfigurable hybrid 3D wireless NoC 360–1
wireless NoCs 345
within-die variation 35, 37, 41
wordline 80, 94
write margin (WM) 80
write voltage margin 130–1, 136

XOR operation 400

York Town Silicon Compiler (YSC) 227

zero-mean Gaussian distribution 15